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. Systematicity

2. Word Order Sensitivity

2.4.2 Tasks

Chapter 3

Understanding semantic generalization through systematicity

Natural Language Understanding (NLU) systems have been extremely successful at reading comprehension tasks, such as question answering (QA) and natural language inference (NLI). These tasks typically test for semantic generalization, where a model has to understand the meaning of the input sentence / passage in order to perform the given task. 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 [Rajpurkar et al., 2016, Nguyen et al., 2016, Trischler et al., 2016, Mostafazadeh et al., 2016, Su et al., 2016], as well as datasets that emphasize commonsense inference, such as entailment between sentences [Bowman et al., 2015b, Williams et al., 2018c].

However, there are growing concerns regarding the ability of NLU systems—and neural networks more generally—to generalize in a systematic and robust way [Bahdanau et al., 2019, Lake and Baroni, 2018, Johnson et al., 2017]. For instance, recent work has highlighted the brittleness of NLU systems to adversarial examples [Jia and Liang, 2017], as well as the fact that NLU models tend to exploit statistical artifacts in datasets, rather than exhibiting true reasoning and generalization capabilities [Gururangan et al., 2018b, Kaushik and Lipton, 2018]. These findings have also dovetailed with the recent dominance of large pre-trained language models, such as BERT, on NLU benchmarks [Devlin et al., 2018, Peters et al., 2018], 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.

In this work, we propose to use the properties of *systematicity* to test the limits of semantic generalization of modern neural networks. As defined by Fodor and Pylyshyn [1988], systematicity test the ability of a system to understand the recombination of known parts and rules. Thus, inspired by the classic AI challenge of inductive logic programming [Quinlan, 1990], 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 [Sinha et al., 2019]. 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 a combination of 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

3.1.1 Notations and Terminology

Following standard practice in formal semantics, we use the term *atom* to refer to a *predicate* symbol and a list of terms, such as $[\texttt{grandfatherOf}, X, Y]$, where the predicate `grandfatherOf` denotes the *relation* between the two *variables*, X and Y. We restrict the predicates to have an arity of 2, i.e., binary predicates. A logical *rule* in this setting is of the form $\mathcal{H} \vdash \mathcal{B}$, where $\mathcal{B}$ is the *body* of the rule, i.e., a conjunction of two *atoms* ($[\alpha_1, \alpha_2]$) and $\mathcal{H}$ is the *head*, i.e., a single atom (α) that can be viewed as the goal or query. For instance, given a knowledge base (KB) R that contains the single rule

$$[\texttt{grandfatherOf}, X, Y] \vdash [[\texttt{fatherOf}, X, Z], [\texttt{fatherOf}, Z, Y]], \tag{3.1}$$

the query $[\texttt{grandfatherOf}, X, Y]$ evaluates to true if and only if the body

$$\mathcal{B} = [[\texttt{fatherOf}, X, Z], [\texttt{fatherOf}, Z, Y]] \tag{3.2}$$

is also true in a given world. A rule is called a *grounded* rule if all atoms in the rule are themselves *grounded*, i.e., all variables are replaced with *constants* or entities in a world. A *fact* is a grounded binary predicate. A *clause* is a conjunction of two or more atoms ($\mathcal{C} = (\mathcal{H}_\mathcal{C} \vdash \mathcal{B}_\mathcal{C} = ([\alpha_1, ..., \alpha_n]))$) which can be built using a set of rules.

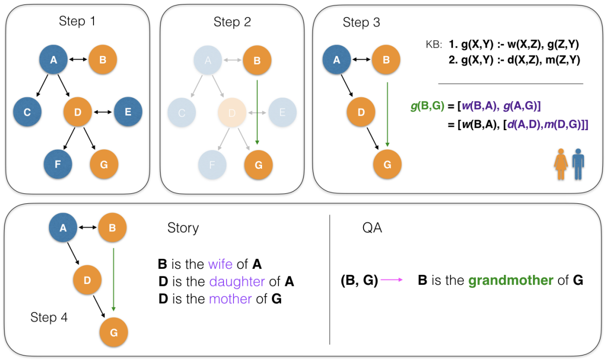

Step 1
A
B
C
D
E
F
G
Step 2
Step 3
KB: 1. g(X,Y) :- w(X,Z), g(Z,Y)
2. g(X,Y) :- d(X,Z), m(Z,Y)
g(B,G) = [w(B,A), g(A,G)]
= [w(B,A), [d(A,D),m(D,G)]]
Step 4
Story
B is the wife of A
D is the daughter of A
D is the mother of G
QA
(B, G)
B is the grandmother of G

Figure 3.1 Data generation pipeline. Step 1: generate a kinship graph. Step 2: sample a target fact. Step 3: Use backward chaining to sample a set of facts. Step 4: Convert sampled facts to a natural language story.

3.2 Overview and construction of CLUTRR

The core idea behind the CLUTRR benchmark suite is the following: Given a natural language story describing a set of kinship relations, the goal is to infer the relationship between two entities, whose relationship is *not* explicitly stated in the story. To generate these stories, we first design a knowledge base (KB) with rules specifying how kinship relations resolve, and we use the following steps to create semi-synthetic stories based on this knowledge base:

Step 1. **Generate** a random kinship graph that satisfies the rules in our KB.

Step 2. **Sample a target fact** (i.e., relation) to predict from the kinship graph.

Step 3. **Apply backward chaining** to sample a set of facts that can prove the target relation (and optionally sample a set of "distracting" or "irrelevant" noise facts).

Step 4. **Convert the sampled facts into a natural language story** through pre-specified text templates and crowd-sourced paraphrasing.

Figure 3.1 provides a high-level overview of this idea, and the following subsections describe the data generation process in detail, as well as the diagnostic flexibility afforded by CLUTRR.

The short stories in CLUTRR are essentially narrativized renderings of a set of logical facts. In the following sections, we describe how we sample the logical facts that make up a story by generating random kinship graphs and using backward chaining to produce logical reasoning chains.

3.2.1 Graph generation

To generate a kinship graph (say, G) underlying a particular story, we first sample a set of gendered[1] entities and kinship relations using a stochastic generation process. This generation process contains a number of tunable parameters—such as the maximum number of children at each node, the probability of an entity being married to another entity, etc.—and is designed to produce a valid, but possibly incomplete "backbone graph". For instance, this backbone graph generation process will specify "parent"/"child" relations between entities but does not add "grandparent" relations. After this initial generation process, we recursively apply the logical rules in R to the backbone graph to produce a final graph G that contains the full set of kinship relations between all the entities. [2]

In the CLUTRR Benchmark, the following kinship relations are used: *son, father, husband, brother, grandson, grandfather, son-in-law, father-in-law, brother-in-law, uncle, nephew, daughter, mother, wife, sister, granddaughter, grandmother, daughter-in-law, mother-in-law, sister-in-law, aunt, niece*.

[1] Kinship and gender roles are oversimplified in our data (compared to the real world) to maintain tractability.

[2] In the context of our data generation process, we distinguish between the knowledge base, R, which contains a finite number of predicates and rules specifying how kinship relations in a family resolve, and a particular kinship graph G, which contains a grounded set of atoms specifying the particular kinship relations that underlie a single story. In other words, R contains the logical rules that govern all the generated stories in CLUTRR, while G contains the grounded facts that underlie a specific story.

$$[\texttt{grand}, X, Y] \vdash [[\texttt{child}, X, Z], [\texttt{child}, Z, Y]],$$

$$[\texttt{grand}, X, Y] \vdash [[\texttt{SO}, X, Z], [\texttt{grand}, Z, Y]],$$

$$[\texttt{grand}, X, Y] \vdash [[\texttt{grand}, X, Z], [\texttt{sibling}, Z, Y]],$$

$$[\texttt{inv-grand}, X, Y] \vdash [[\texttt{inv-child}, X, Z], [\texttt{inv-child}, Z, Y]],$$

$$[\texttt{inv-grand}, X, Y] \vdash [[\texttt{sibling}, X, Z], [\texttt{inv-grand}, Z, Y]],$$

$$[\texttt{child}, X, Y] \vdash [[\texttt{child}, X, Z], [\texttt{sibling}, Z, Y]],$$

$$[\texttt{child}, X, Y] \vdash [[\texttt{SO}, X, Z], [\texttt{child}, Z, Y]],$$

$$[\texttt{inv-child}, X, Y] \vdash [[\texttt{sibling}, X, Z], [\texttt{inv-child}, Z, Y]],$$

$$[\texttt{inv-child}, X, Y] \vdash [[\texttt{child}, X, Z], [\texttt{inv-grand}, Z, Y]],$$

$$[\texttt{sibling}, X, Y] \vdash [[\texttt{child}, X, Z], [\texttt{inv-un}, Z, Y]],$$

$$[\texttt{sibling}, X, Y] \vdash [[\texttt{inv-child}, X, Z], [\texttt{child}, Z, Y]]$$

$$[\texttt{sibling}, X, Y] \vdash [[\texttt{sibling}, X, Z], [\texttt{sibling}, Z, Y]],$$

$$[\texttt{in-law}, X, Y] \vdash [[\texttt{child}, X, Z], [\texttt{SO}, Z, Y]],$$

$$[\texttt{inv-in-law}, X, Y] \vdash [[\texttt{SO}, X, Z], [\texttt{inv-child}, Z, Y]],$$

$$[\texttt{un}, X, Y] \vdash [[\texttt{sibling}, X, Z], [\texttt{child}, Z, Y]],$$

$$[\texttt{inv-un}, X, Y] \vdash [[\texttt{inv-child}, X, Z], [\texttt{sibling}, Z, Y]],$$

We used a small, tractable, and logically sound KB of rules as mentioned above. We carefully select this set of deterministic rules to avoid ambiguity in the resolution. We use gender-neutral predicates and resolve the gender of the predicate in the head $\mathcal{H}$ of a clause $\mathcal{C}$ by deducing the gender of the second constant. We have two types of predicates, *vertical* predicates (parent-child relations) and *horizontal* predicates (sibling or significant other). We denote all the vertical predicates by its *child-to-parent* relation and append the prefix `inv-` to the predicates for the corresponding *parent-to-child* relation. For example, `grandfatherOf` is denoted by the gender-neutral predicate $[\texttt{inv-grand}, X, Y]$, where the gender is determined by the gender of Y.

3.2.2 Backward chaining

The resulting graph G provides the *background knowledge* for a specific story, as each edge in this graph can be treated as a grounded predicate (i.e., fact) between two entities. From this graph G, we sample the facts that make up the story, as well as the target fact that we seek to predict: First, we (uniformly) sample a target relation $\mathcal{H}_{\mathcal{C}}$, which is the fact that we want to predict from the story. Then, from this target relation $\mathcal{H}_{\mathcal{C}}$, we run a simple variation of the backward chaining [Gallaire and Minker, 1978] algorithm for k iterations starting from $\mathcal{H}_{\mathcal{C}}$, where at each iteration we uniformly sample a subgoal to resolve and then uniformly sample a KB rule that resolves this subgoal. Crucially, unlike traditional backward chaining, we do not stop the algorithm when a proof is obtained; instead, we run for a fixed number of iterations k in order to sample a set of k facts $\mathcal{B}_{\mathcal{C}}$ that imply the target relation $\mathcal{H}_{\mathcal{C}}$.

3.2.3 Adding natural language

So far, we have described the process of generating a conjunctive logical clause $\mathcal{C} = (\mathcal{H}_{\mathcal{C}} \vdash \mathcal{B}_{\mathcal{C}})$, where $\mathcal{H}_{\mathcal{C}} = [\alpha^*]$ is the target fact (i.e., relation) we seek to predict and $\mathcal{B}_{\mathcal{C}} = [\alpha_1, ..., \alpha_k]$ is the set of supporting facts that imply the target relation. We now describe how we convert this logical representation to natural language through crowd-sourcing.

Paraphrasing using Amazon Mechanical Turk

We use Amazon Mechanical Turk (AMT), an online platform for collecting annotations from crowd-workers [3]. The platform supports a mechanism to quickly annotate large amounts of data by paying anonymous workers for their effort. In our work, the crowd-workers are shown a set of facts $\mathcal{B}_{\mathcal{C}}$ corresponding to a story and then they

[3] https://www.mturk.com/

are asked to paraphrase these facts into a narrative. Since workers are given a set of facts $\mathcal{B}_C$ to work from, they are able to combine and split multiple facts across separate sentences and construct diverse narratives (Figure 3.3).

We use ParlAI [Miller et al., 2017] Mturk interface to collect paraphrases from the users. Specifically, given a set of facts, we ask the users to paraphrase the facts into a story. The users (*turkers*) are free to construct any story they like as long as they mention all the entities and all the relations among them. We also provide the head $\mathcal{H}$ of the clause as an *inferred* relation and specifically instruct the users to *not* mention it in the paraphrased story. In order to evaluate the paraphrased stories, we ask the turkers to peer review a story paraphrased by a different turker. Since there are two tasks - paraphrasing a story and rating a story - we choose to pay 0.5$ for each annotation. A sample task description in our MTurk interface is as follows:

> In this task, you will need to write a short, simple story based on a few facts. **It is crucial that the story mentions each of the given facts at least once.** The story does not need to be complicated! It just needs to be grammatical and mention the required facts.
>
> After writing the story, you will be asked to evaluate the quality of a generated story (based on a different set of facts). **It is crucial that you check whether the generated story mentions each of the required facts.**
>
> *Example of good and bad stories: Good Example*
>
> **Facts to Mention**
>
> - John is the father of Sylvia.
> - Sylvia has a brother Patrick.
>
> **Implied Fact**: John is the father of Patrick.
>
> **Written story**

John is the proud father of the lovely Sylvia. Sylvia has a love-hate relationship with her brother Patrick.

Bad Example

Facts to Mention

- Vincent is the son of Tim.
- Martha is the wife of Tim.

Implied Fact : Martha is Vincent's mother.

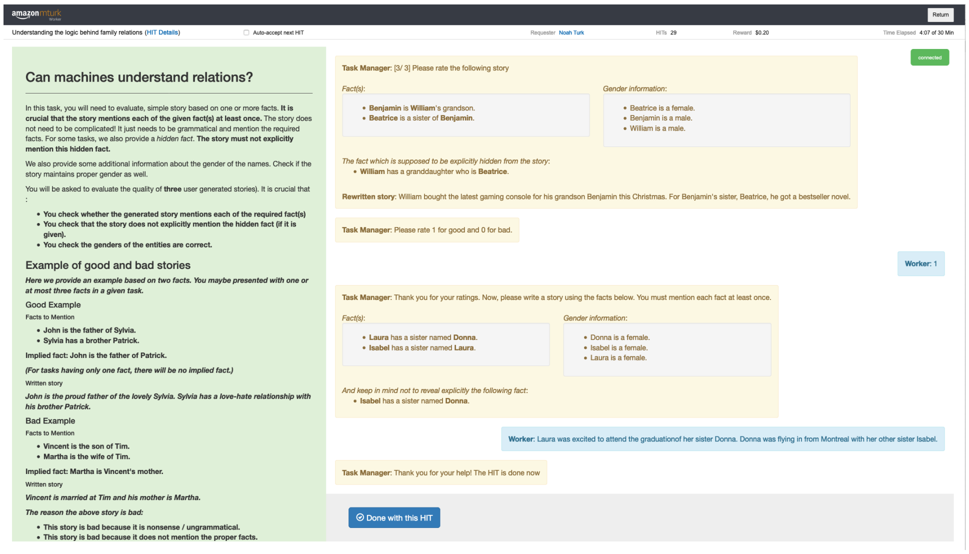
Written story

Vincent is married at Tim and his mother is Martha.

The reason the above story is bad:

- This story is bad because it is nonsense / ungrammatical.
- This story is bad because it does not mention the proper facts.
- This story is bad because it reveals the implied fact.

A sample of the AMT interface is shown in Figure 3.2. To ensure that the turkers are providing high-quality annotations without revealing the inferred fact, we also launch another task to ask the turkers to rate three annotations to be either good or bad which are provided by a set of *different* turkers. We pay 0.2$ for each HIT consisting of three reviews. This helped to remove logical and grammatical inconsistencies to a large extent. Based on the reviews, 79% of the collected paraphrases passed the peer-review sanity check where all the reviewers agree on the quality. This subset of the placeholders is used in the benchmark. A sample of programmatically generated dataset for clause length of $k = 2$ to $k = 6$ is provided in the tables 3.3 to 3.7.



Understanding the logic behind family relations (HIT Details) □ Auto-accept next HIT Requester Noah Turk HITs 29 Reward $0.20 Time Elapsed 4:07 of 30 Min

Can machines understand relations?

In this task, you will need to evaluate, simple story based on one or more facts. **It is crucial that the story mentions each of the given fact(s) at least once.** The story does not need to be complicated! It just needs to be grammatical and mention the required facts. For some tasks, we also provide a *hidden fact*. **The story must not explicitly mention this hidden fact.**

We also provide some additional information about the gender of the names. Check if the story maintains proper gender as well.

You will be asked to evaluate the quality of **three** user generated stories). It is crucial that :

- **You check whether the generated story mentions each of the required fact(s)**
- **You check that the story does not explicitly mention the hidden fact (if it is given).**
- **You check the genders of the entities are correct.**

Example of good and bad stories

Here we provide an example based on two facts. You maybe presented with one or at most three facts in a given task.

Good Example

Facts to Mention

- **John is the father of Sylvia.**
- **Sylvia has a brother Patrick.**

Implied fact: John is the father of Patrick.

(For tasks having only one fact, there will be no implied fact.)

Written story

John is the proud father of the lovely Sylvia. Sylvia has a love-hate relationship with his brother Patrick.

Bad Example

Facts to Mention

- **Vincent is the son of Tim.**
- **Martha is the wife of Tim.**

Implied fact: Martha is Vincent's mother.

Written story

Vincent is married at Tim and his mother is Martha.

The reason the above story is bad:

- **This story is bad because it is nonsense / ungrammatical.**
- **This story is bad because it does not mention the proper facts.**

Task Manager: [3/ 3] Please rate the following story

Fact(s):

- **Benjamin** is **William**'s grandson.
- **Beatrice** is a sister of **Benjamin**.

Gender information:

- Beatrice is a female.
- Benjamin is a male.
- William is a male.

The fact which is supposed to be explicitly hidden from the story:

- **William** has a granddaughter who is **Beatrice**.

Rewritten story: William bought the latest gaming console for his grandson Benjamin this Christmas. For Benjamin's sister, Beatrice, he got a bestseller novel.

Task Manager: Please rate 1 for good and 0 for bad.

Worker: 1

Task Manager: Thank you for your ratings. Now, please write a story using the facts below. You must mention each fact at least once.

Fact(s):

- **Laura** has a sister named **Donna**.
- **Isabel** has a sister named **Laura**.

Gender information:

- Donna is a female.
- Isabel is a female.
- Laura is a female.

And keep in mind not to reveal explicitly the following fact:

- **Isabel** has a sister named **Donna**.

Worker: Laura was excited to attend the graduationof her sister Donna. Donna was flying in from Montreal with her other sister Isabel.

Task Manager: Thank you for your help! The HIT is done now

Figure 3.2 Amazon Mechanical Turker Interface built using ParlAI which was used to collect data as well as peer reviews.

Reusability and composition

One challenge for data collection via AMT is that the number of possible stories generated by CLUTRR grows combinatorially as the number of supporting facts increases, i.e., as $k = |\mathcal{B}_\mathcal{C}|$ grows. This combinatorial explosion for large k—combined with the difficulty of maintaining the quality of the crowd-sourced paraphrasing for long stories—makes it infeasible to obtain a large number of paraphrased examples for $k > 3$. To circumvent this issue and increase the flexibility of our benchmark, we reuse and compose AMT paraphrases to generate longer stories. In particular, we collected paraphrases for stories containing $k = 1, 2, 3$ supporting facts and then replaced the entities from these collected stories with placeholders in order to re-use them to generate longer semi-synthetic stories. An example of a story generated by stitching together two shorter paraphrases is provided below:

> [Frank] went to the park with his father, [Brett]. [Frank] called his brother [Boyd] on the phone. He wanted to go out for some beers. [Boyd] went to the baseball game with his son [Jim].
>
> Q: What is [Brett] and [Jim]'s relationship?

Thus, instead of simply collecting paraphrases for a fixed number of stories, we instead obtain a diverse collection of natural language templates that can be programmatically recombined to generate stories with various properties.

3.2.4 AMT Template statistics

Number of Paraphrases			# clauses
	k = 1	1,868	20
	k = 2	1,890	58
	k = 3	2,258	236
	Total	6,016	
Unique Word Count		3,797	
Jaccard Word Overlap	Unigrams	0.201	
	Bigrams	0.0385	

Table 3.1 Statistics of the AMT paraphrases. Jaccard word overlap is calculated within the templates of each individual clause of length k.

At the time of submission, we have collected 6,016 unique paraphrases with an average of 19 paraphrases for every possible logical clause of length $k = 1, 2, 3$. Table 3.1 contains summary statistics of the collected paraphrases. Overall, we found high linguistic diversity in the collected paraphrases. For instance, the average Jaccard overlap in unigrams between pairs paraphrases corresponding to the same logical clause was only 0.201 and only 0.0385 for bigrams.

3.2.5 Human performance

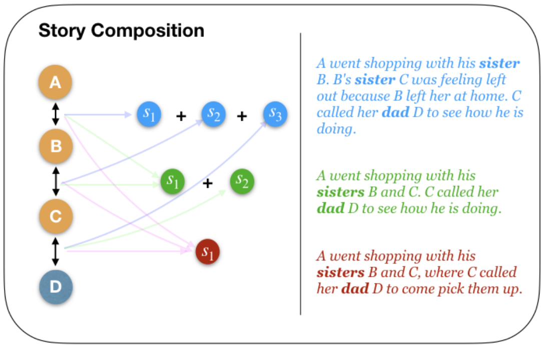
To get a sense of the data quality and difficulty involved in CLUTRR, we asked human annotators to solve the task for random examples of length $k = 2, 3, ..., 6$. (Table 3.2)

Relation Length	Human Performance Time Limited	 Unlimited Time	Reported Difficulty
2	0.848	1	1.488 +- 1.25
3	0.773	1	2.41 +- 1.33
4	0.477	1	3.81 +- 1.46
5	0.424	1	3.78 +- 0.96
6	0.406	1	4.46 +- 0.87

Table 3.2 Human performance accuracies on CLUTRR dataset. Humans are provided the Clean-Generalization version of the dataset, and we test on two scenarios: when a human is given limited time to solve the task, and when a human is given unlimited time to solve the task. Regardless of time, our evaluators provide a score of difficulty of individual puzzles.

We perform the evaluation in two scenarios: first a time-limited scenario where we ask AMT Turkers to solve the puzzle in a fixed time. Turkers were provided a maximum time of 30 mins, but they solved the puzzles in an average of 1 minute 23 seconds. Secondly, we use another set of expert evaluators who are given ample time to solve the tasks. Not surprisingly, if a human being is given ample time (experts took an average of 6 minutes per puzzle) and a pen and a paper to aid in the reasoning, they get all the relations correct. However, if an evaluator is short of time, they might miss important details on the relations and perform poorly. Thus, our tasks require *active attention*.

We found that time-constrained AMT annotators performed well (i.e., $> 70\%$) accuracy for $k \leq 3$ but struggled with examples involving longer stories, achieving 40-50% accuracy for $k > 3$. However, trained annotators with unlimited time were able to solve 100% of the examples, highlighting the fact that this task requires attention and involved reasoning, even for humans.



*A went shopping with his **sister** B. B's **sister** C was feeling left out because B left her at home. C called her **dad** D to see how he is doing.*

*A went shopping with his **sisters** B and C. C called her **dad** D to see how he is doing.*

*A went shopping with his **sisters** B and C, where C called her **dad** D to come pick them up.*

Figure 3.3 Illustration of how a set of facts can split and combined in various ways across sentences.

3.2.6 Representing the question and entities

The AMT paraphrasing approach described above allows us to convert the set of supporting facts $\mathcal{B}_\mathcal{C}$ to a natural language story, which can be used to predict the target relation/query $\mathcal{H}_\mathcal{C}$. However, instead of converting the target query, $\mathcal{H}_\mathcal{C} = [\alpha^*]$, to a natural language question, we instead opt to represent the target query as a K-way classification task, where the two entities in the target relation are provided as input and the goal is to classify the relation that holds between these two entities. This representation avoids the pitfall of revealing information about the answer in the question [Kaushik and Lipton, 2018].

When generating stories, entity names are randomly drawn from a set of 300 common gendered English names. Thus, depending on each run, the entities are never the same. This ensures that the entity names are simply placeholders and uncorrelated from the task.

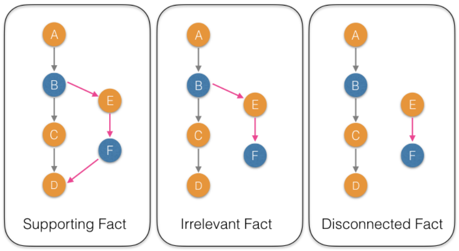


Figure 3.4 Noise generation procedures of CLUTRR.

3.3 Experimental Setups

The modular nature of CLUTRR provides rich diagnostic capabilities for evaluating the robustness and generalization abilities of neural language understanding systems. We highlight some key diagnostic capabilities available via different variations of CLUTRR below. These diagnostic variations correspond to the concrete datasets that we generated in this work, and we describe the results on these datasets in §3.5.

3.3.1 Systematic generalization

Most prominently, CLUTRR allows us to explicitly evaluate a model's ability for generalizing with the property of systematicity. In particular, we rely on the following hold-out procedures to test systematic generalization:

- During training, we hold out a subset of the collected paraphrases, and we only use this held-out subset of paraphrases when generating the test set. Thus, to succeed on CLUTRR, an NLU system must exhibit *linguistic generalization* and be robust to linguistic variation at test time.
- We also hold out a subset of the logical clauses during training (for clauses of length $k > 2$).[4] In other words, during training, the model sees all logical rules but does not

[4] One should not holdout clauses from length $k = 2$ in order to allow models to learn the compositionality of all possible binary predicates.

see all *combinations* of these logical rules. Thus, in addition to linguistic generalization, success on this task also requires *logical generalization*.

- Lastly, as a more extreme form of both logical and linguistic generalization, we consider the setting where the models are trained on stories generated from clauses of length $\leq k$ and evaluated on stories generated from larger clauses of length $> k$. Thus, we explicitly test the ability for models to generalize on examples that require more steps of reasoning that any example they encountered during training.

3.3.2 Robust Reasoning

In addition to evaluating systematic generalization, the modular setup of CLUTRR also allows us to diagnose model robustness by adding *noise facts* to the generated narratives. Due to the controlled semi-synthetic nature of CLUTRR, we are able to provide a precise taxonomy of the kinds of noise facts that can be added (Figure 3.4). In order to structure this taxonomy, it is important to recall that any set of supporting facts $\mathcal{B}_C$ generated by CLUTRR can be interpreted as a path, p_C, in the corresponding kinship graph G (Figure 3.1). Based on this interpretation, we view adding noise facts from the perspective of sampling three different types of noise paths, p_n, from the kinship graph G:

- *Irrelevant facts*: We add a path p_n, which has exactly one shared end-point with p_c. In this way, this is a *distractor* path, which contains facts that are connected to one of the entities in the target relation, $\mathcal{H}_C$, but do not provide any information that could be used to help answer the query.
- *Supporting facts*: We add a path p_n, whose two end-points are on the path p_C. The facts on this path p_n are noise because they are not needed to answer the query, but they are supporting facts because they can, in principle, be used to construct alternative (longer) reasoning paths that connect the two target entities.
- *Disconnected facts*: We add paths which neither originate nor end in any entity on p_c.

These disconnected facts involve entities and relations that are completely unrelated to the target query.

3.3.3 Generated Datasets

For all experiments, we generated datasets with 10-15k training examples. In many experiments, we report training and testing results on stories with different clause lengths k. (For brevity, we use the phrase "clause length" throughout this section to refer to the value $k = |\mathcal{B}_{\mathcal{C}}|$, i.e., the number of steps of reasoning that are required to predict the target query.) In all cases, the training set contains 5000 train stories per k value, and, during testing, all experiments use 100 test stories per k value. All experiments were run 10 times with different randomly generated stories, and means and standard errors over these 10 runs are reported. As discussed above, during training we holdout 20% of the paraphrases, as well as 10% of the possible logical clauses.

Table 3.3 Snapshot of puzzles in the dataset for k=2

Puzzle	**Question**	**Gender**	**Answer**
Charles's son *Christopher* entered rehab for the ninth time at the age of thirty. *Randolph* had a nephew called *Christopher* who had n't seen for a number of years.	Randolph is the _____ of Charles	Charles:male, Christopher:male, Randolph:male	brother
Randolph and his sister *Sharon* went to the park. *Arthur* went to the baseball. game with his son *Randolph*	Sharon is the _____ of Arthur	Arthur:male, Randolph:male, Sharon:female	daughter
Frank went to the park with his father, *Brett*. *Frank* called his brother *Boyd* on the phone. He wanted to go out for some beers.	Brett is the _____ of Boyd	Boyd:male, Frank:male, Brett:male	father

3.4 Evaluated Models

Our primary baselines are neural language understanding models that take unstructured text as input. We consider bidirectional LSTMs [Hochreiter and Schmidhuber, 1997, Cho et al., 2014] (with and without attention), as well as models that aim to incorporate inductive biases towards relational reasoning: Relation Networks (RN) [Santoro

Table 3.4 Snapshot of puzzles in the dataset for k=3

Puzzle	Question	Gender	Answer
Roger was playing baseball with his sons *Sam* and *Leon*. *Sam* had to take a break though because he needed to call his sister *Robin*.	Leon is the _____ of Robin	Robin:female, Sam:male, Roger:male, Leon:male	brother
Elvira and her daughter *Nancy* went shopping together last Monday and they bought new shoes for *Elvira*'s kids. *Pedro* and his sister *Allison* went to the fair. *Pedro*'s mother, *Nancy*, was out with friends for the day.	Elvira is the _____ of Allison	Allison:female, Pedro:male, Nancy:female, Elvira:female	grandmother
Roger met up with his sister *Nancy* and her daughter *Cynthia* at the mall to go shopping together. *Cynthia*'s brother *Pedro* was going to be the star in the new show.	Pedro is the _____ of Roger	Roger:male, Nancy:female, Cynthia:female, Pedro:male	nephew

Table 3.5 Snapshot of puzzles in the dataset for k=4

Puzzle	Question	Gender	Answer
Celina has been visiting her sister, *Fran* all week. *Fran* is also the daughter of *Bethany*. *Ronald* loves visiting his aunt *Bethany* over the weekends. *Samuel*'s son *Ronald* entered rehab for the ninth time at the age of thirty.	Celina is the _____ of Samuel	Samuel:male, Ronald:male, Bethany:female, Fran:female, Celina:female	niece
Celina adores her daughter *Bethany*. *Bethany* loves her very much, too. *Jackie* called her mother *Bethany* to let her know she will be back home soon. *Thomas* was helping his daughter *Fran* with her homework at home. Afterwards, *Fran* and her sister *Jackie* played Xbox together.	Celina is the _____ of Thomas	Thomas:male, Fran:female, Jackie:female, Bethany:female, Celina:female	daughter
Raquel is *Samuel* 'daughter and they go shopping at least twice a week together. *Kenneth* and her mom, *Theresa*, had a big fight. *Theresa*'s son, *Ronald*, refused to get involved. *Ronald* was having an argument with her sister, *Raquel*.	Samuel is the _____ of Kenneth	Kenneth:male, Theresa:female, Ronald:male, Raquel:female, Samuel:male	father

Table 3.6 Snapshot of puzzles in the dataset for k=5

Puzzle	Question	Gender	Answer
Steven's son is *Bradford*. *Bradford* and his father always go fishing together on Sundays and have a great time together. *Diane* is taking her brother *Brad* out for a late dinner. *Kristin*, *Brad*'s mother, is home with a cold. *Diane*'s father *Elmer*, and his brother *Steven*, all got into the rental car to start the long cross-country roadtrip they had been planning.	Bradford is the _____ of Kristin	Kristin:female, Brad:male, Diane:female, Elmer:male, Steven:male, Bradford:male	nephew
Elmer went on a roadtrip with his youngest child, *Brad*. *Lena* and her sister *Diane* are going to a restaurant for lunch. *Lena*'s brother *Brad* is going to meet them there with his father *Elmer* *Brad* ca n't stand his unfriendly aunt *Lizzie*.	Lizzie is the _____ of Diane	Diane:female, Lena:female, Brad:male, Elmer:male, Lizzie:female	aunt
Ira took his niece *April* fishing Saturday. They caught a couple small fish. *Ronald* was enjoying spending time with his parents, *Damion* and *Claudine*. *Damion*'s other son, *Dennis*, wanted to come visit too. *Dennis* often goes out for lunch with his sister, *April*.	Ira is the _____ of Claudine	Claudine:female, Ronald:male, Damion:male, Dennis:male, April:female, Ira:male	brother

Table 3.7 Snapshot of puzzles in the dataset for k=6

Puzzle	**Question**	**Gender**	**Answer**
Mario wanted to get a good gift for his sister, *Marianne*. *Jean* and her sister *Darlene* were going to a party held by *Jean*'s mom, *Marianne*. *Darlene* invited her brother *Roy* to come, too, but he was too busy. *Teri* and her father, *Mario*, had an argument over the weekend. However, they made up by Monday. *Agnes* wants to make a special meal for her daughter *Teri*'s birthday.	Roy is the _____ of Agnes	Agnes:female, Teri:female, Mario:male, Marianne:female, Jean:female, Darlene:female, Roy:male	nephew
Robert's aunt, *Marianne*, asked *Robert* to mow the lawn for her. *Robert* said he could n't because he had a bad back. *William*'s parents, *Brian* and *Marianne*, threw him a surprise party for his birthday. *Brian*'s daughter *Jean* made a mental note to be out of town for her birthday! *Agnes*'s biggest accomplishment is raising her son *Robert*. *Jean* is looking for a good gift for her sister *Darlene*.	Darlene is the _____ of Agnes	Agnes:female, Robert:male, Marianne:female, William:male, Brian:male, Jean:female, Darlene:female	niece
Sharon and her brother *Mario* went shopping. *Teri*, *Mario*'s daughter, came too. *Agnes*, *Annie*'s mother, is unhappy with *Robert*. She feels her son is cruel to *Annie*'s sister *Teri*, and she wants *Robert* to be nicer. *Robert*'s sister, *Nicole*, participated in the dance contest.	Nicole is the _____ of Sharon	Sharon:female, Mario:male, Teri:female, Annie:female, Agnes:female, Robert:male, Nicole:female	niece

et al., 2017], Relational Recurrent Networks (RMC) [Santoro et al., 2018] and Compositional Memory Attention Network (MAC) [Hudson and Manning, 2018]. We also use the large pre-trained language model, BERT [Devlin et al., 2018], as well as a modified version of BERT having a trainable LSTM encoder on top of the pretrained BERT embeddings. All of these models (except BERT) were re-implemented in PyTorch 1.0 [Paszke et al., 2017] and adapted to work with the CLUTRR benchmark.

Since the underlying relations in the stories generated by CLUTRR inherently form a graph, we also experiment with a Graph Attention Network (GAT) [Veličković et al., 2018]. Rather than taking the textual stories as input, the GAT baseline receives a structured graph representation of the facts that underlie the story.

Entity and query representations. We use the various baseline models to encode the natural language story (or graph) into a fixed-dimensional embedding. With the exception of the BERT models, we do not use pre-trained word embeddings and learn the word embeddings from scratch using end-to-end backpropagation. An important note, however, is that we perform Cloze-style anonymization [Hermann et al., 2015] of the entities (i.e., names) in the stories, where each entity name is replaced by a *@entity-k* placeholder, which is randomly sampled from a small, fixed pool of placeholder tokens. The embeddings for these placeholders are randomly initialized and fixed during training.

To make a prediction about a target query given a story, we concatenate the embedding of the story (generated by the baseline model) with the embeddings of the two target entities and we feed this concatenated embedding to a 2-layer feed-forward neural network with a softmax prediction layer.

3.4.1 Hyperparameters

For all models, the common hyperparameters used are: Embedding dimension: 100 (except BERT based models), Optimizer: Adam, Learning rate: 0.001, Number of epochs: 100, Number of runs: 10. Specific model-based hyperparameters are given as follows:

- **Bidirectional LSTM** [Hochreiter and Schmidhuber, 1997, Cho et al., 2014]: LSTM hidden dimension: 100, # layers: 2, Classifier MLP hidden dimension: 200
- **Relation Networks** [Santoro et al., 2017]: f_{θ_1} : 256, f_{θ_2}: 64, g_θ : 64
- **Compositional Memory Attention Network (MAC)** [Hudson and Manning, 2018]: # Iterations: 6, `shareQuestion`: True, Dropout - Memory, Read and Write: 0.2
- **Relational Recurrent Networks** [Santoro et al., 2018]: Memory slots: 2, Head size: 192, Number of heads: 4, Number of blocks : 1, forget bias : 1, input bias: 0, gate style: unit, key size: 64, # Attention layers: 3, Dropout: 0
- **BERT** [Devlin et al., 2018]: Layers : 12, Fixed pretrained embeddings from `bert-base-uncased` using Pytorch HuggingFace BERT repository [5], Word dimension: 768, appended with a two-layer MLP for final prediction.
- **BERT-LSTM**: Same parameters as above, with a two-layer unidirectional LSTM encoder on top of BERT word embeddings.
- **GAT** [Veličković et al., 2018]: Node dimension: 100, Message dimension: 100, Edge dimension: 20, number of rounds: 3

3.5 Results

We evaluate several NLU systems on the proposed CLUTRR benchmark to surface the relative strengths and shortcomings of these models in the context of inductive reasoning and combinatorial generalization.[6] We aim to answer the following key questions:

[5] https://github.com/huggingface/pytorch-pretrained-BERT

[6] Code to reproduce all the results in this section are available at https://github.com/facebookresearch/clutrr/.

(**Q1**) How do state-of-the-art NLU models compare in terms of systematic generalization? Can these models generalize to stories with unseen combinations of logical rules?

(**Q2**) How does the performance of neural language understanding models compare to a graph neural network that has full access to graph structure underlying the stories?

(**Q3**) How robust are these models to the addition of noise facts to a given story?

3.5.1 Systematic Generalization

We begin by using CLUTRR to evaluate the ability of the baseline models to perform systematic generalization (**Q1**). In this setting, we consider two training regimes: in the first regime, we train all models with clauses of length $k = 2, 3$, and in the second regime, we train with clauses of length $k = 2, 3, 4$. We then test the generalization of these models on test clauses of length $k = 2, ..., 10$.

Figure 3.5 illustrates the performance of different models on this generalization task. We observe that the GAT model is able to perform near-perfectly on the held-out logical clauses of length $k = 3$, with the BERT-LSTM being the top-performer among the text-based models but still significantly below the GAT. Not surprisingly, the performance of all models degrades monotonically as we increase the length of the test clauses, which highlights the challenge of "zero-shot" systematic generalization Lake and Baroni [2018], Sodhani et al. [2018]. However, as expected, all models improve on their generalization performance when trained on $k = 2, 3, 4$ rather than just $k = 2, 3$ (Figure 3.5, right). The GAT, in particular, achieves the biggest gain by this expanded training.

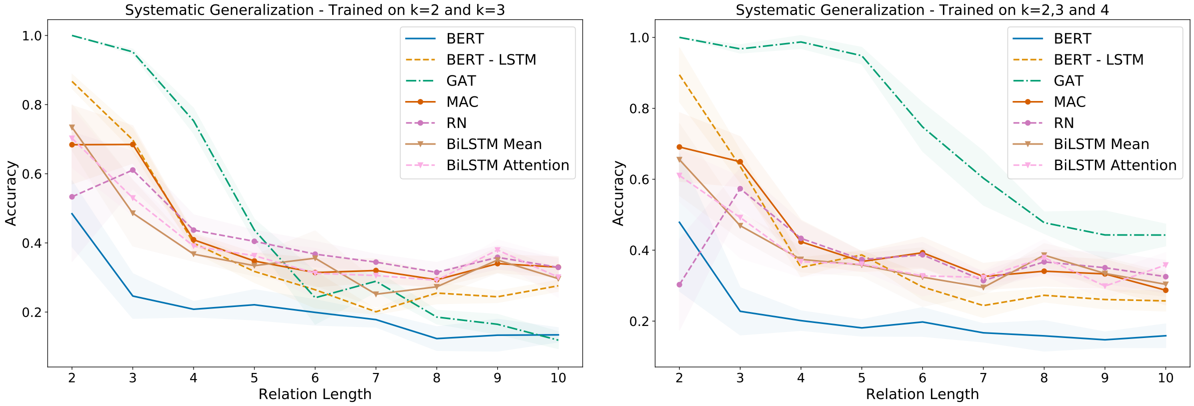


Figure 3.5 Systematic generalization performance of different models when trained on clauses of length $k = 2, 3$ (Left) and $k = 2, 3, 4$ (Right).

3.5.2 The benefit of structure

The empirical results on systematic generalization also provide insight into how the text-based NLU systems compare against the graph-based GAT model that has full access to the logical graph structure underlying the stories (**Q2**). Indeed, the relatively strong performance of the GAT model (Figure 3.5) suggests that the language-based models fail to learn a robust mapping from the natural language narratives to the underlying logical facts.

To further confirm this trend, we ran experiments with modified train and test splits for the text-based models, where the same set of natural language paraphrases were used to construct the narratives in both the train and test splits (Figure 3.6). In this simplified setting, the text-based models must still learn to reason about held-out logical patterns, but the difficulty of parsing the natural language is essentially removed, as the same natural language paraphrases are used during testing and training. We found that the text-based models were competitive with the GAT model in this simplified setting, confirming that the poor performance of the text-based models on the main task is driven by the difficulty of parsing the unseen natural language narratives.

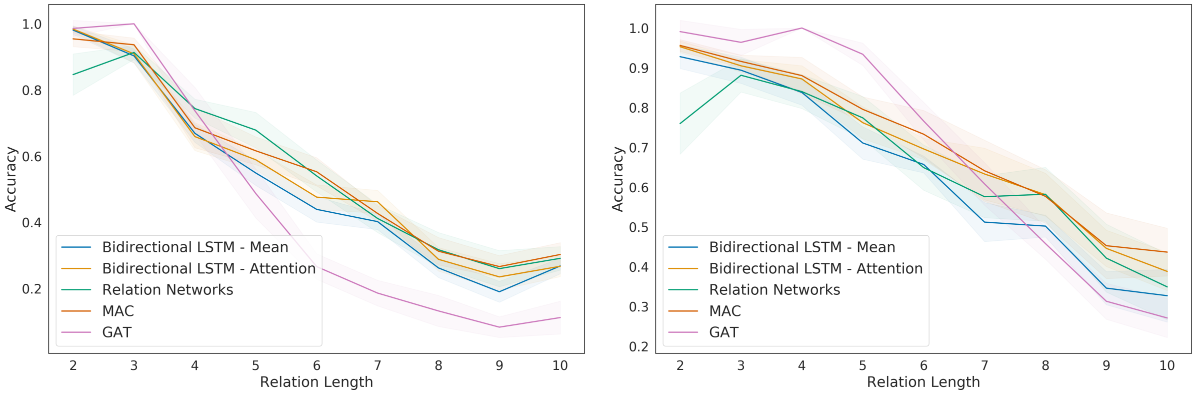


Figure 3.6 Systematic Generalizability of different models on `CLUTRR-Gen` task (having 20% less placeholders and without training and testing placeholder split), when **Left:** trained with $k = 2$ and $k = 3$ and **Right:** trained with $k = 2, 3$ and 4

Models		Unstructured models (no graph)						Structured model (with graph)
Training	Testing	BiLSTM - Attention	BiLSTM - Mean	RN	MAC	BERT	BERT-LSTM	GAT
Clean	Clean	0.58 ±0.05	0.53 ±0.05	0.49 ±0.06	0.63 ±0.08	0.37 ±0.06	0.67 ±0.03	**1.0** ±0.0
	Supporting	**0.76** ±0.02	0.64 ±0.22	0.58 ±0.06	0.71 ±0.07	0.28 ±0.1	0.66 ±0.06	0.24 ±0.2
	Irrelevant	0.7 ±0.15	**0.76** ±0.02	0.59 ±0.06	0.69 ±0.05	0.24 ±0.08	0.55 ±0.03	0.51 ±0.15
	Disconnected	0.49 ±0.05	0.45 ±0.05	0.5 ±0.06	0.59 ±0.05	0.24 ±0.08	0.5 ±0.06	**0.8** ±0.17
Supporting	Supporting	0.67 ±0.06	0.66 ±0.07	0.68 ±0.05	0.65 ±0.04	0.32 ±0.09	0.57 ±0.04	**0.98** ±0.01
Irrelevant	Irrelevant	0.51 ±0.06	0.52 ±0.06	0.5 ±0.04	0.56 ±0.04	0.25 ±0.06	0.53 ±0.06	**0.93** ±0.01
Disconnected	Disconnected	0.57 ±0.07	0.57 ±0.06	0.45 ±0.11	0.4 ±0.1	0.17 ±0.05	0.47 ±0.06	**0.96** ±0.01
Average		**0.61** ±0.08	0.59 ±0.08	0.54 ±0.07	**0.61** ±0.06	0.30 ±0.07	0.56 ±0.05	**0.77** ±0.09

Table 3.8 Testing the robustness of the various models when training and testing on stories containing various types of noise facts. The types of noise facts (supporting, irrelevant, and disconnected) are defined in Section .

3.5.3 Robust Reasoning

Finally, we use CLUTRR to systematically evaluate how various baseline neural language understanding systems cope with noise (**Q3**). In all the experiments we provide a combination of $k = 2$ and $k = 3$ length clauses in training and testing, with noise facts being added to the train and/or test set depending on the setting (Table 3.8). We use the different types of noise facts defined in Section 3.3.2..

Overall, we find that the GAT baseline outperforms the unstructured text-based models across most testing scenarios (Table 3.8), which showcases the benefit of a structured feature space for robust reasoning. When training on clean data and testing on noisy data, we observe two interesting trends that highlight the benefits and shortcomings of the various model classes:

1. All the text-based models excluding BERT actually perform better when testing on examples that have *supporting* or *irrelevant* facts added. This suggests that these models actually benefit from having more content related to the entities in the story. Even though this content is not strictly useful or needed for the reasoning task, it may provide some linguistic cues (e.g., about entity genders) that the models exploit. In contrast, the BERT-based models do not benefit from the inclusion of this extra content, which is perhaps due to the fact that they are already built upon a strong language model (e.g., that already adequately captures entity genders.)

2. The GAT model performs poorly when *supporting* facts are added but has no performance drop when *disconnected* facts are added. This suggests that the GAT model is sensitive to changes that introduce cycles in the underlying graph structure but is robust to the addition of noise that is disconnected from the target entities.

Learning from noisy data

Moreover, when we trained on noisy examples, we found that only the GAT model was able to consistently improve its performance (Table 3.8). We notice that the GAT model, having access to the true underlying graph of the puzzles, perform better across different testing scenarios when trained with the noisy data. As the *Supporting facts* contains cycles, it is difficult for GAT to generalize for a dataset with cycles when it is trained on a dataset without cycles. However, when trained with cycles, GAT learns to attend to *all* the paths leading to the correct answer. This effect is disastrous when GAT is tested on *Irrelevant facts* which contains dangling paths as GAT still tries to attend to all the paths. Training on *Irrelevant facts* proved to be most beneficial to GAT, as the

Models		Unstructured models (no graph)						Structured model (with graph)
Training	Testing	BiLSTM - Attention	BiLSTM - Mean	RN	MAC	BERT	BERT-LSTM	GAT
Supporting	Clean	0.38 ±0.04	0.32 ±0.04	0.4 ±0.09	0.45 ±0.03	0.19 ±0.06	0.39 ±0.06	**0.92** ±0.17
	Supporting	0.67 ±0.06	0.66 ±0.07	0.68 ±0.05	0.65 ±0.04	0.32 ±0.09	0.57 ±0.04	**0.98** ±0.01
	Irrelevant	0.44 ±0.03	0.39 ±0.03	**0.51** ±0.08	0.46 ±0.09	0.2 ±0.06	0.36 ±0.05	0.5 ±0.23
	Disconnected	0.31 ±0.21	0.25 ±0.16	0.47 ±0.08	0.41 ±0.06	0.2 ±0.08	0.32 ±0.04	**0.92** ±0.05
Irrelevant	Clean	0.57 ±0.05	0.56 ±0.05	0.46 ±0.13	0.67 ±0.05	0.24 ±0.06	0.46 ±0.08	**0.92** ±0.0
	Supporting	0.38 ±0.22	0.31 ±0.16	0.61 ±0.07	0.61 ±0.04	0.27 ±0.06	0.46 ±0.04	**0.77** ±0.12
	Irrelevant	0.51 ±0.06	0.52 ±0.06	0.5 ±0.04	0.56 ±0.04	0.25 ±0.06	0.53 ±0.06	**0.93** ±0.01
	Disconnected	0.44 ±0.26	0.54 ±0.27	0.55 ±0.05	0.61 ±0.06	0.26 ±0.03	0.45 ±0.08	**0.85** ±0.25
Disconnected	Clean	0.45 ±0.02	0.47 ±0.03	0.53 ±0.09	0.5 ±0.06	0.22 ±0.09	0.44 ±0.05	**0.75** ±0.07
	Supporting	0.47 ±0.03	0.46 ±0.05	0.54 ±0.03	0.58 ±0.06	0.22 ±0.06	0.38 ±0.08	**0.78** ±0.12
	Irrelevant	0.47 ±0.05	0.48 ±0.03	0.52 ±0.04	0.51 ±0.05	0.17 ±0.04	0.38 ±0.05	**0.56** ±0.26
	Disconnected	0.57 ±0.07	0.57 ±0.06	0.45 ±0.11	0.4 ±0.1	0.17 ±0.05	0.47 ±0.06	**0.96** ±0.01
Average		0.47 ±0.08	0.46 ±0.08	0.52 ±0.07	**0.53** ±0.06	0.23 ±0.07	0.43 ±0.05	**0.82** ±0.09

Table 3.9 Testing the robustness of the various models when trained various types of noisy facts and evaluated on other noisy / clean facts.

model now perfectly attends to *only relevant paths*. Since *Disconnected facts* contains disconnected paths, the message passing function of the graph is unable to forward any information from the disjoint cliques, thereby having superior testing scores throughout several scenarios.

Again, these results highlights the performance gap between the unstructured text-based models and GAT for solving the CLUTRR task.

Learning with synthetic placeholders

In order to further understand the effect of language placeholders on robustness, we performed another set of experiments where we use bABI Weston et al. [2015] style simple placeholders (Table 3.10). We observe a marked increase in performance of all NLU models, where they significantly decrease the gap between their performance with that of GAT, even outperforming GAT on various settings. This shows the significance of using paraphrased placeholders in devising the complexity of the dataset.

Models		Unstructured models (no graph)						Structured model (with graph)
Training	Testing	BiLSTM - Attention	BiLSTM - Mean	RN	MAC	BERT	BERT-LSTM	GAT
Supporting	Clean	0.96 ±0.01	**0.97** ±0.01	0.88 ±0.05	0.94 ±0.02	0.48 ±0.08	0.57 ±0.08	0.92 ±0.17
	Supporting	0.96 ±0.03	0.96 ±0.03	0.97 ±0.01	0.97 ±0.01	0.75 ±0.07	0.88 ±0.05	**0.98** ±0.01
	Irrelevant	0.92 ±0.02	**0.93** ±0.01	0.9 ±0.03	0.91 ±0.01	0.56 ±0.04	0.54 ±0.06	0.5 ±0.23
	Disconnected	0.8 ±0.04	0.83 ±0.04	0.76 ±0.08	0.86 ±0.04	0.27 ±0.06	0.42 ±0.08	**0.92** ±0.05
Irrelevant	Clean	0.63 ±0.02	0.61 ±0.07	0.85 ±0.09	0.8 ±0.07	0.53 ±0.09	0.44 ±0.06	**0.92** ±0.0
	Supporting	0.66 ±0.03	0.64 ±0.04	0.69 ±0.06	0.76 ±0.06	0.42 ±0.08	0.43 ±0.08	**0.77** ±0.12
	Irrelevant	0.89 ±0.04	0.86 ±0.1	0.74 ±0.11	0.78 ±0.06	0.61 ±0.1	0.83 ±0.06	**0.93** ±0.01
	Disconnected	0.64 ±0.02	0.62 ±0.05	0.72 ±0.05	0.73 ±0.04	0.41 ±0.04	0.61 ±0.05	**0.85** ±0.25
Disconnected	Clean	0.9 ±0.05	0.82 ±0.12	**0.94** ±0.02	0.93 ±0.04	0.68 ±0.07	0.64 ±0.02	0.75 ±0.07
	Supporting	0.87 ±0.04	0.82 ±0.05	0.85 ±0.03	**0.88** ±0.04	0.54 ±0.08	0.5 ±0.05	0.78 ±0.12
	Irrelevant	**0.87** ±0.03	0.85 ±0.03	0.83 ±0.03	0.87 ±0.02	0.59 ±0.09	0.58 ±0.09	0.56 ±0.26
	Disconnected	0.91 ±0.04	0.91 ±0.03	0.8 ±0.17	0.71 ±0.11	0.49 ±0.1	0.79 ±0.1	**0.96** ±0.01
Average		0.83 ±0.08	0.82 ±0.08	0.83 ±0.07	**0.84** ±0.06	0.58 ±0.07	0.60 ±0.05	**0.82** ±0.09

Table 3.10 Testing the robustness on toy placeholders of the various models when trained various types of noisy facts and evaluated on other noisy / clean facts.

3.6 Related Work

To design the CLUTRR dataset, we draw inspiration from the classic work on inductive logic programming (ILP), a long line of reading comprehension benchmarks in NLP, as well as work combining language and knowledge graphs.

3.6.1 Reading comprehension benchmarks

Many datasets have been proposed to test the reading comprehension ability of NLP systems. This includes the SQuAD Rajpurkar et al. [2016], NewsQA Trischler et al. [2016], and MCTest Richardson et al. [2013] benchmarks that focus on factual questions; the SNLI Bowman et al. [2015b] and MultiNLI Williams et al. [2018c] benchmarks for sentence understanding; and the bABI tasks Weston et al. [2015], to name a few. Our primary contribution to this line of work is the development of a carefully designed *diagnostic* benchmark to evaluate model robustness and systematic generalization in the context of NLU.

3.6.2 Systematic generalization

A growing body of literature has demonstrated that NLU models tend to exploit statistical artifacts in datasets and lack true generalization capabilities Jia and Liang [2017], Gururangan et al. [2018b], Kaushik and Lipton [2018], Lake and Baroni [2018]. These critical examinations have dovetailed with similar studies on visual question answering [Agrawal et al., 2016, Bahdanau et al., 2019, Johnson et al., 2017]. CLUTRR, contributes to this growing area by introducing a principled and flexible benchmark to evaluate systematic generalization in the context of language understanding—with our notion of systematic generalization being grounded in classic work on inductive logic programming (ILP) Quinlan [1990].

3.6.3 Question-answering with knowledge graphs

Our work is also related to the domain of question answering and reasoning in knowledge graphs [Das et al., 2018, Xiong et al., 2018, Hamilton et al., 2018, Wang et al., 2018, Xiong et al., 2017, Welbl et al., 2018, Kartsaklis et al., 2018], where either the model is provided with a knowledge graph to perform inference over or where the model must infer a knowledge graph from the text itself. However, unlike previous benchmarks in this domain—which are generally *transductive* and focus on leveraging and extracting knowledge graphs as a source of background knowledge about a fixed set of entities—CLUTRR requires *inductive logical reasoning*, where every example requires reasoning over a new set of previously unseen entities.

3.7 Discussion

In this paper we introduced the CLUTRR benchmark suite to test the systematic generalization and inductive reasoning capabilities of NLU systems. We demonstrated the diagnostic capabilities of CLUTRR and found that existing NLU systems exhibit relatively poor robustness and systematic generalization capabilities—especially when compared to a graph neural network that works directly with symbolic input. Concretely, using CLUTRR we were able to make the following key insights about the reasoning capability of modern neural networks:

- **Neural language models are unable to reason when tested with systematicity.** We saw in §3.5.1 that the performance of all NLU models drastically degrade when we test on instances which require systematicity - the knowledge of combination of existing parts - to solve the task. While all models had access to all possible rules (by ingesting a combination of relations in the training data), all models are notably worse when tested with longer chain of reasoning than the ones trained upon. This shortcoming could be due to overly associating to certain patterns seen during training, or learning to solve the task by taking shortcuts - associating some combination of tokens for certain relations [Gururangan et al., 2018b].

- **Models are not robust in their language understanding.** When evaluated with enabling (supporting) and distractor information (noise), we observe models to display conflicting results. While supporting information is indeed useful for certain classes of models (§3.5.3), irrelevant and distracting information also seems to aide in the reasoning process, which is not a systematic behaviour. Furthermore, when trained with noise, majority of the NLU models are unable to discern between the correct and the incorrect information. These results indicate a potential surface form realization issue.

- **The key hurdle behind systematic generalization is the natural language itself.** Finally, we observe overwhelmingly that when a model which is only provided a graph, stripped of the natural language layer, the model is able to reason with surprising ability. The graph model, GAT, does not have to extract the relevant information from a given free-form text. This makes it easier for the model to generalize more effectively, even in the scenarios when the model is tasked to learn from distractor (noisy) information.

These results highlight the gap that remains between machine reasoning models that work with unstructured text and models that are given access to more structured input. It appears the key hindrance for a neural model for effective generalization and reasoning is the access to proper surface forms. These results raises questions on the syntax processing capabilities of NLU models, and call for more in-depth investigation on the same. In fact, in the following chapters of this thesis, I will discuss my works on further studying the notions of syntax encoding in NLU models using the tool of systematicity.

3.8 Follow-up findings in the community

Chapter 4

Quantifying syntactic generalization using word order

Of late, large scale pre-trained Transformer-based [Vaswani et al., 2017] models—such as RoBERTa [Liu et al., 2019], BART [Lewis et al., 2020], and GPT-2 and -3 [Radford et al., 2019, Brown et al., 2020]—have exceeded recurrent neural networks' performance on many NLU tasks [Wang et al., 2018, 2019]. Several papers have even suggested that Transformers pretrained on a language modeling (LM) objective can capture syntactic information [Hewitt and Manning, 2019, Jawahar et al., 2019, Warstadt and Bowman, 2020, Wu et al., 2020], with their self-attention layers being capable of surprisingly effective learning Rogers et al. [2020]. In the preceeding chapter, we observed that NLU models, including BERT, are unable to reason systematicity, primarily due to their lack of understanding the surface forms of the given task. Thus, in this chapter, we question the claim that state-of-the-art NLU models "know syntax".

Since there are many ways to investigate "syntax", we must be clear on what we mean by the term. Knowing the syntax of a sentence means being sensitive to the *order of the words* in that sentence (among other things). Humans are sensitive to word order, so clearly, "language is not merely a bag of words" [Harris, 1954, p.156]. Moreover, it is easier for us to identify or recall words presented in canonical orders than in disordered, ungrammatical sentences; this phenomenon is called the *"sentence superiority effect"* (Cattell 1886, Scheerer 1981, Toyota 2001, Baddeley et al. 2009, Snell and Grainger 2017, 2019, Wen et al. 2019, i.a.). This effect also finds some neurobiological support from work showing ordered text activates portions of the temporal lobe more than unordered word lists [Bemis and Pylkkänen, 2013, Pylkkänen et al., 2014]. In our estimation then, if one wants to claim that a model "knows syntax", then they should minimally show that the model is sensitive to word order (at least for e.g. English or Mandarin Chinese).

Generally, knowing the syntax of a sentence is taken to be a prerequisite for understanding what that sentence means [Heim and Kratzer, 1998]. Models should have to know the syntax first then, if performing any particular NLU task that genuinely requires a humanlike understanding of meaning (cf. Bender and Koller 2020). Thus, if our models are as good at NLU as our current evaluation methods suggest, we should expect them to be sensitive to word order. In this chapter, I discuss our paper Sinha et al. [2021b] where we use a suite of permutation metrics to find the models are not sensitive to word order.

We focus here on textual entailment, one of the hallmark tasks used to measure how well models understand language [Condoravdi et al., 2003, Dagan et al., 2005]. This task, often also called Natural Language Inference (NLI; Bowman et al. 2015a, i.a.), typically consists of two sentences: a premise and a hypothesis. The objective is to predict whether the premise entails the hypothesis, contradicts it, or is neutral with respect to it. We find rampant word order insensitivity in purportedly high performing NLI models. For nearly all premise-hypothesis pairs, **there are many permuted examples that fool the models** into providing the correct prediction. In case of MNLI, for example, the current state-of-the-art of 90.5% can be increased to **98.7**% merely by permuting the word order of test set examples. We even find drastically increased cross-dataset

Premise	**Hypothesis**	**Predicted Label**
Boats in daily use lie within feet of the fashionable bars and restaurants.	There are boats close to bars and restaurants.	E
restaurants and use feet of fashionable lie the in Boats within bars daily .	bars restaurants are There and to close boats .	E
He and his associates weren't operating at the level of metaphor.	He and his associates were operating at the level of the metaphor.	C
his at and metaphor the of were He operating associates n't level .	his the and metaphor level the were He at associates operating of .	C

Table 4.1 Examples from the MNLI Matched development set. Both the original example and the permuted one elicit the same classification label (entailment and contradiction respectively) from RoBERTa (large). A simple demo is provided in an associated Google Colab notebook.

generalization when we reorder words. This is not just a matter of chance—we show that the model output probabilities are significantly different from uniform. A sample of the model outputs with permuted examples is shown in Table 4.1.

We verify our findings with three popular English NLI datasets—SNLI [Bowman et al., 2015a], MultiNLI [Williams et al., 2018b] and ANLI [Nie et al., 2020])—and one Chinese one, OCNLI Hu et al. [2020a]. It is thus less likely that our findings result from some quirk of English or a particular tokenization strategy. We also observe the effect for various transformer architectures pre-trained on language modeling (RoBERTa [Liu et al., 2019], BART [Lewis et al., 2020], DistilBERT [Sanh et al., 2020]), and non-transformers, including a ConvNet [Zhao et al., 2015], an InferSent model [Conneau et al., 2017], and a BiLSTM [Collobert and Weston, 2008].

Thus, in this chapter I discuss our contributions in Sinha et al. [2021b], which are as follows: (i) we propose a suite of metrics (*Permutation Acceptance*) for measuring model insensitivity to word order (§4.2), (ii) we construct multiple permuted test datasets for measuring NLI model performance at a large scale (§4.4), (iii) we show that NLI models focus on words more than word order, but can partially reconstruct syntactic information from words alone (§4.5.1), (iv) we show the problem persists on out-of-domain data, (v) we show that humans struggle with UnNatural Language Inference, underscoring the non-humanlikeness of SOTA models (§4.5.2), (vi) finally, we explore a simple maximum entropy-based method (§4.5.3) to encourage models not to accept permuted examples.

4.1 Technical Background

4.2 Experimental Setup

4.2.1 Constructing the permuted dataset.

For a given dataset D having splits D_{train} and D_{test}, we first train an NLI model M on D_{train} to achieve comparable accuracy to what was reported in the original papers. We then construct a randomized version of D_{test}, which we term as $\hat{D}_{\text{test}}$ such that: for each example $(p_i, h_i, y_i) \in D_{\text{test}}$ (where p_i and h_i are the premise and hypothesis sentences of the example respectively and y_i is the gold label), we use a permutation operator $\mathcal{F}$ that returns a list ($\hat{P}_i, \hat{H}_i$) of q permuted sentences ($\hat{p}_i$ and $\hat{h}_i$), where q is a hyperparameter. $\mathcal{F}$ essentially permutes all positions of the words in a given sentence (i.e., either in premise or hypothesis) with the restriction that *no words maintain their original position*. In our initial setting, we do not explicitly control the placement of the words relative to their original neighbors, but we analyze clumping effects in §4.4. $\hat{D}_{\text{test}}$ now consists of

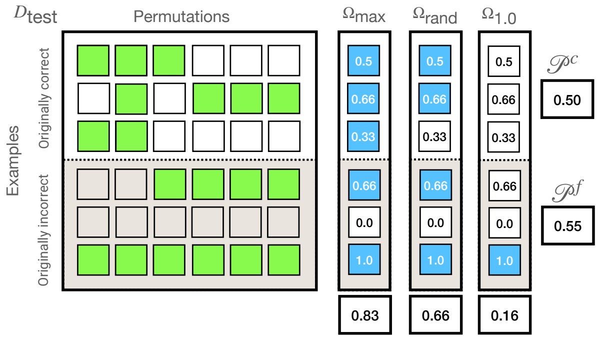


Figure 4.1 Graphical representation of the Permutation Acceptance class of metrics. Given a sample test set D_{test} with six examples, three of which originally predicted correctly (model predicts gold label), three incorrectly (model fails to predict gold label), with $n = 6$ permutations, Ω_{max},Ω_{rand}, $\Omega_{1.0}$, $\mathcal{P}^c$ and $\mathcal{P}^f$ are provided. Green boxes indicate permutations accepted by the model. Blue boxes mark examples that crossed each threshold and were used to compute the corresponding metric.

$|D_{\text{test}}| \times q$ examples, with q different permutations of hypothesis and premise for each original test example pair. If a sentence S (e.g., h_i) contains w words, then the total number of available permutations of S are $(w-1)!$, thus making the output of $\mathcal{F}$ a list of $\binom{(w-1)!}{q}$ permutations in this case. For us, the space of possible outputs is larger, since we permute p_i and h_i separately (and ignore examples for which any $|S| \leq 5$).

4.2.2 Defining Permutation Acceptance.

The choice of q naturally allows us to analyze a statistical view of the predictability of a model on the permuted sentences. To that end, we define the following notational conventions. Let $\mathcal{A}$ be the original accuracy of a given model M on a dataset D, and c be the number of examples in a dataset which are marked as correct according to the standard formulation of accuracy for the original dataset (i.e., they are assigned the ground truth label). Typically $\mathcal{A}$ is given by $\frac{c}{|D_{test}|}$ or $\frac{c}{|D_{dev}|}$.

Let $\Pr_M(\hat{P}_i, \hat{H}_i)_{\text{cor}}$ then be the percentage of q permutations of an example (p_i, h_i)

assigned the ground truth label y_i by M:

$$\Pr_M(\hat{P}_i, \hat{H}_i)_{\text{cor}} = \frac{1}{q} \sum_{(\hat{p}_j \in \hat{P}_i, \hat{h}_j \in \hat{H}_i)} ((M(\hat{p}_j, \hat{h}_j) = y_i) \to 1) \tag{4.1}$$

To get an overall summary score, we let Ω_x be the percentage of examples $(p_i, h_i) \in D_{\text{test}}$ for which $\Pr_M(\hat{P}_i, \hat{H}_i)_{\text{cor}}$ exceeds a predetermined threshold $0 < x < 1$. Concretely, a given example will count as correct according to Ω_x if more than x percent of its permutations ($\hat{P}_i$ and $\hat{H}_i$) are assigned y_i by the model M. Mathematically,

$$\Omega_x = \frac{1}{\mid D_{test} \mid} \sum_{(p_i, h_i) \in D_{test}} ((\Pr_M(\hat{P}_i, \hat{H}_i)_{\text{cor}} > x) \to 1). \tag{4.2}$$

There are two specific cases of Ω_{x} that we are most interested in. First, we define Ω_{max} or the **Maximum Accuracy**, where $x = 1/|D_{\text{test}}|$. In short, Ω_{max} gives the percentage of examples $(p_i, h_i) \in D_{\text{test}}$ for which there is *at least one* permutation $(\hat{p_j}, \hat{h_j})$ that model M assigns the gold label y_i [1]. Second, we define Ω_{rand}, or **Random Baseline Accuracy**, where $x = 1/m$ or chance probability (for balanced m-way classification, where $m = 3$ in NLI). This metric is less stringent than Ω_{max}, as it counts an example if at least *one third* of its permutations are assigned the gold label (hence provides a lower-bound relaxation). See Figure 4.1 for a graphical representation of Ω_{x}.

We also define D^f to be the list of examples originally marked incorrect according to $\mathcal{A}$, but are now deemed correct according Ω_{max}. D^c is the list of examples originally marked correct according to $\mathcal{A}$. Thus, we should expect $D^f < D^c$ for models that have high accuracy. Additionally, we define $\mathcal{P}^c$ and $\mathcal{P}^f$, as the dataset average percentage of permutations which predicted the gold label, when the examples were originally correct (D^c) and when the examples were originally incorrect (D^f) as per $\mathcal{A}$ (hence,

[1] Theoretically, $\Omega_{\text{max}} \to 1$ if the number of permutations q is large. Thus, in our experiments we set $q = 100$.

flipped) respectively.

$$\mathcal{P}^c = \frac{1}{|D^c|} \sum_{i=0}^{|D^c|} M(\hat{P}_i, \hat{H}_i)_{\text{cor}} \tag{4.3}$$

P^f is defined similarly by replacing D^c by D^f. Note that for a classic BOW model, $\mathcal{P}^c = 100$ and $\mathcal{P}^f = 0$, because it would rely on the words alone (not their order) to make its classification decision. Since permuting removes no words, BOW models should come to the same decisions for permuted examples as for the originals.

4.3 Evaluated Models

We run our experiments on two types of models: **(a)** Transformer-based models and **(b)** Non-Transformer Models. In **(a)**, we investigate the state-of-the-art pre-trained models such as RoBERTa-Large Liu et al. [2019], BART-Large Lewis et al. [2020] and DistilBERT Sanh et al. [2020]. For **(b)** we consider several recurrent and convolution based neural networks, such as InferSent Conneau et al. [2017], Bidirectional LSTM Collobert and Weston [2008] and ConvNet Zhao et al. [2015]. We train all models on MNLI, and evaluate on in-distribution (SNLI and MNLI) and out-of-distribution datasets (ANLI). We independently verify results of **(a)** using both our fine-tuned model using HuggingFace Transformers Wolf et al. [2020] and pre-trained checkpoints from FairSeq Ott et al. [2019] (using PyTorch Model Hub). For **(b)**, we use the InferSent codebase. We sample $q = 100$ permutations for each example in D_{test}, and use 100 seeds for each of those permutations to ensure full reproducibility. We drop examples from test sets where we are unable to compute *all unique* randomizations, typically these are examples with sentences of length of less than 6 tokens. [2]

[2] Code, data, and model checkpoints are available at https://github.com/facebookresearch/unlu.

4.4 Results

4.4.1 Models accept many permuted examples.

We find Ω_{max} is very high for models trained and evaluated on MNLI (in-domain generalization), reaching **98.7%** on MNLI dev. and test sets (in RoBERTa, compared to $\mathcal{A}$ of 90.6% (Table 4.2). Recall, human accuracy is approximately 92% on MNLI dev., Nangia and Bowman 2019). This shows that there exists at least one permutation (usually many more) for almost all examples in D_{test} such that model M predicts the gold label. We also observe high Ω_{rand} at 79.4%, showing that there are many examples for which the models outperform even a random baseline in accepting permuted sentences (see **??** for more Ω values.) We provide an example of the behaviour in Table 4.1.

Evaluating out-of-domain generalization with ANLI dataset splits resulted in an Ω_{max} value that is notably higher than $\mathcal{A}$ (89.7% Ω_{max} for RoBERTa compared to 45.6% $\mathcal{A}$). As a consequence, we encounter many *flips*, i.e., examples where the model is unable to predict the gold label, but at least one permutation of that example is able to. However, recall this analysis expects us to know the gold label upfront, so this test can be thought of as running a word-order probe test on the model until the model predicts the gold label (or give up by exhausting our set of q permutations). For out-of-domain generalization, Ω_{rand} decreases considerably (36.4% Ω_{rand} on A1), which means fewer permutations are accepted by the model. Next, recall that a classic bag-of-words model would have $\mathcal{P}^c = 100$ and $\mathcal{P}^f = 0$. No model performs strictly like a classic bag of words although they do perform somewhat BOW-like ($\mathcal{P}^c >> \mathcal{P}^f$ for all test splits, **??**). We find this BOW-likeness to be higher for certain non-Transformer models, (InferSent) as they exhibit higher $\mathcal{P}^c$ (84.2% for InferSent compared to 70.7% for RoBERTa on MNLI).

Model	Eval. Dataset	$\mathcal{A}$	Ω_{max}	$\mathcal{P}^c$	$\mathcal{P}^f$	Ω_{rand}
RoBERTa-Large	MNLI_m_dev	0.906	0.987	0.707	0.383	0.794
	MNLI_mm_dev	0.901	0.987	0.707	0.387	0.790
	SNLI_dev	0.879	0.988	0.768	0.393	0.826
	SNLI_test	0.883	0.988	0.760	0.407	0.828
	A1*	0.456	0.897	0.392	0.286	0.364
	A2*	0.271	0.889	0.465	0.292	0.359
	A3*	0.268	0.902	0.480	0.308	0.397
	Mean	0.652	0.948	0.611	**0.351**	0.623
BART-Large	MNLI_m_dev	0.902	0.989	0.689	0.393	0.784
	MNLI_mm_dev	0.900	0.986	0.695	0.399	0.788
	SNLI_dev	0.886	0.991	0.762	0.363	0.834
	SNLI_test	0.888	0.990	0.762	0.370	0.836
	A1*	0.455	0.894	0.379	0.295	0.374
	A2*	0.316	0.887	0.428	0.303	0.397
	A3*	0.327	0.931	0.428	0.333	0.424
	Mean	**0.668**	**0.953**	0.592	**0.351**	**0.634**
DistilBERT	MNLI_m_dev	0.800	0.968	0.775	0.343	0.779
	MNLI_mm_dev	0.811	0.968	0.775	0.346	0.786
	SNLI_dev	0.732	0.956	0.767	0.307	0.731
	SNLI_test	0.738	0.950	0.770	0.312	0.725
	A1*	0.251	0.750	0.511	0.267	0.300
	A2*	0.300	0.760	0.619	0.265	0.343
	A3*	0.312	0.830	0.559	0.259	0.363
	Mean	0.564	0.883	**0.682**	0.300	0.575
InferSent	MNLI_m_dev	0.658	0.904	0.842	0.359	0.712
	MNLI_mm_dev	0.669	0.905	0.844	0.368	0.723
	SNLI_dev	0.556	0.820	0.821	0.323	0.587
	SNLI_test	0.560	0.826	0.824	0.321	0.600
	A1*	0.316	0.669	0.425	0.395	0.313
	A2*	0.310	0.662	0.689	0.249	0.330
	A3*	0.300	0.677	0.675	0.236	0.332
	Mean	**0.481**	0.780	0.731	**0.322**	0.514
ConvNet	MNLI_m_dev	0.631	0.926	0.773	0.340	0.684
	MNLI_mm_dev	0.640	0.926	0.782	0.343	0.694
	SNLI_dev	0.506	0.819	0.813	0.339	0.597
	SNLI_test	0.501	0.821	0.809	0.341	0.596
	A1*	0.271	0.708	0.648	0.218	0.316
	A2*	0.307	0.725	0.703	0.224	0.356
	A3*	0.306	0.798	0.688	0.234	0.388
	Mean	0.452	**0.817**	**0.745**	0.291	0.519
BiLSTM	MNLI_m_dev	0.662	0.925	0.800	0.351	0.711
	MNLI_mm_dev	0.681	0.924	0.809	0.344	0.724
	SNLI_dev	0.547	0.860	0.762	0.351	0.598
	SNLI_test	0.552	0.862	0.771	0.363	0.607
	A1*	0.262	0.671	0.648	0.271	0.340
	A2*	0.297	0.728	0.672	0.209	0.328
	A3*	0.304	0.731	0.656	0.219	0.331
	Mean	0.472	0.814	0.731	0.301	**0.520**

Table 4.2 Statistics for Transformer-based models trained on MNLI corpus Williams et al. [2018b]. The highest values are bolded (**red** indicates the model most insensitive to permutation) per metric and per model class (Transformers and non-Transformers). A1*, A2* and A3* refer to the ANLI dev. sets [Nie et al., 2020].

Model	$\mathcal{A}$	Ω_{max}	$\mathcal{P}^c$	$\mathcal{P}^f$	Ω_{rand}
RoBERTa-Large	**0.784**	**0.988**	0.726	**0.339**	**0.773**
InferSent	0.573	0.931	0.771	0.265	0.615
ConvNet	0.407	0.752	**0.808**	0.199	0.426
BiLSTM	0.566	0.963	0.701	0.271	0.611

Table 4.3 Results on evaluation on OCNLI Dev set. All models are trained on OCNLI corpus Hu et al. [2020a]. Bold marks the highest value per metric (**red** shows the model is insensitive to permutation).

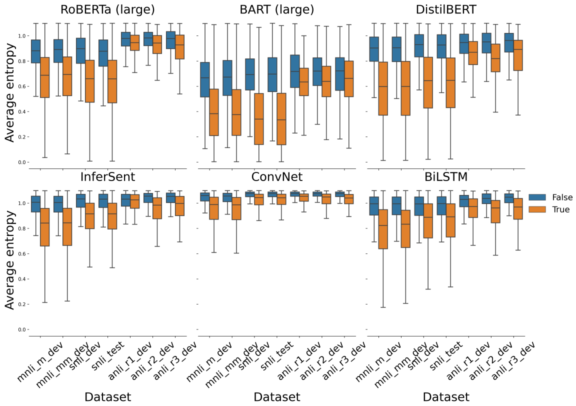


Figure 4.2 Average entropy of model confidences on permutations that yielded the correct results for Transformer-based models (top) and Non-Transformer-based models (bottom). Results are shown for D^c (orange) and D^f (blue). The boxes show the quartiles of the entropy distributions.

4.4.2 Models are very confident.

The phenomenon we observe would be of less concern if the correct label prediction was just an outcome of chance, which could occur when the entropy of the log probabilities of the model output is high (suggesting uniform probabilities on entailment, neutral and contradiction labels, recall Model B from **??**). We first investigate the model probabilities for the Transformer-based models on the permutations that lead to the correct answer in Figure 4.2. We find overwhelming evidence that model confidences on in-distribution datasets (MNLI, SNLI) are highly skewed, resulting in low entropy, and it varies among different model types. BART proves to be the most skewed Transformer-based model. This skewness is not a property of model capacity, as we observe DistilBERT log probabilities to have similar skewness as RoBERTa (large) model, while exhibiting lower $\mathcal{A}$, Ω_{max}, and Ω_{rand}.

For non-Transformers whose accuracy $\mathcal{A}$ is lower, the Ω_{max} achieved by these models are also predictably lower. We observe roughly the same relative performance in the terms of Ω_{max} (**??** and Appendix Table 4.2) and Average entropy (Figure 4.2). However, while comparing the averaged entropy of the model predictions, it is clear that there is some benefit to being a worse model—non-Transformer models are not as overconfident on randomized sentences as Transformers are. High confidence of Transformer models can be attributed to the *overthinking* phenomenon commonly observed in deep neural networks Kaya et al. [2019] and BERT-based models Zhou et al. [2020].

4.4.3 Similar artifacts in Chinese NLU.

We extended the experiments to the Original Chinese NLI dataset [Hu et al., 2020a, OCNLI], and re-used the pre-trained RoBERTa-Large and InferSent (non-Transformer) models on OCNLI. Our findings are similar to the English results (Table 4.3), thereby suggesting that the phenomenon is not just an artifact of English text or tokenization.

4.4.4 Other Results.

We investigated the effect of sentence length (which correlates with number of possible permutations; **??**), and hypothesis-only randomization (models exhibit similar phenomenon even when only hypothesis is permuted; **??**).

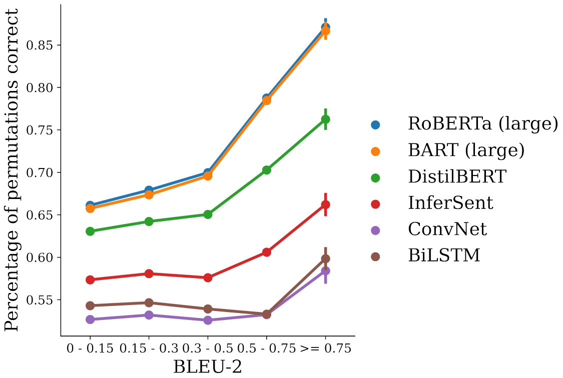


Figure 4.3 BLEU-2 score versus acceptability of permuted sentences across all test datasets. RoBERTa and BART performance is similar but differs considerably from the performance of non-Transformer-based models, such as InferSent and ConvNet.

4.5 Analysis

4.5.1 Analyzing Syntactic Structure Associated with Tokens

A natural question to ask following our findings: what is it about particular permutations that leads models to accept them? Since the permutation operation is drastic and only rarely preserves local word relations, we first investigate whether there exists a relationship between Permutation Acceptance scores and local word order preservation. Concretely, we compare bi-gram word overlap (BLEU-2) with the percentage of permutations that are deemed correct (Figure 4.3).[3] Although the probability of a permuted sentence to be predicted correctly does appear to track BLEU-2 score (Figure 4.3), the percentage of examples which were assigned the gold label by the Transformer-based models is still higher than we would expect from permutations

[3] We observe, due to our permutation process, the maximum BLEU-3 and BLEU-4 scores are negligibly low (< 0.2 BLEU-3 and < 0.1 BLEU-4), already calling into question the hypothesis that n-grams are the sole explanation for our finding. Because of this, we only compare BLEU-2 scores. Detailed experiments on specially constructed permutations that cover the entire range of BLEU-3 and BLEU-4 is provided in **??**.

with lower BLEU-2 (66% for the lowest BLEU-2 range of $0 - 0.15$), suggesting preserved relative word order alone cannot explain the high permutation acceptance rates.

Thus, we find that local order preservation does correlate with Permutation Acceptance, but it doesn't fully explain the high Permutation Acceptance scores. We now further ask whether Ω is related to a more abstract measure of local word relations, i.e., part-of-speech (POS) neighborhood.

Many syntactic formalisms, like Lexical Functional Grammar [Kaplan and Bresnan, 1995, Bresnan et al., 2015, LFG], Head-drive Phrase Structure Grammar [Pollard and Sag, 1994, HPSG] or Lexicalized Tree Adjoining Grammar [Schabes et al., 1988, Abeille, 1990, LTAG], are "lexicalized", i.e., individual words or morphemes bear syntactic features telling us which other words they can combine with. For example, "buy" could be associated with (at least) two lexicalized syntactic structures, one containing two noun phrases (as in *Kim bought cheese*), and another with three (as in *Lee bought Logan cheese*). We speculate that our NLI models might accept permuted examples at high rates, because they are (perhaps noisily) reconstructing the original sentence from abstract, word-anchored information about common neighbors.

To test this, we POS-tagged D_{train} using 17 Universal Part-of-Speech tags (using spaCy, Honnibal et al. 2020). For each $w_i \in S_i$, we compute the occurrence probability of POS tags on tokens in the *neighborhood* of w_i. The neighborhood is specified by the radius r (a symmetrical window r tokens from $w_i \in S_i$ to the left and right). We denote this sentence level probability of neighbor POS tags for a word w_i as $\psi^r_{\{w_i,S_i\}} \in \mathcal{R}^{17}$ (see an example in **??** in the Appendix). Sentence-level word POS neighbor scores can be averaged across D_{train} to get a type level score $\psi^r_{\{w_i,D_{\text{train}}\}} \in \mathcal{R}^{17}, \forall w_i \in D_{\text{train}}$. Then, for a sentence $S_i \in D_{\text{test}}$, for each word $w_i \in S_i$, we compute a **POS mini-tree overlap score**:

$$\begin{aligned}\beta^k_{\{w_i,S_i\}} = \frac{1}{k} \mid \text{argmax}_k \psi^r_{\{w_i,D_{\text{train}}\}} \cap \\ \text{argmax}_k \psi^r_{\{w_i,S_i\}} \mid\end{aligned} \tag{4.4}$$

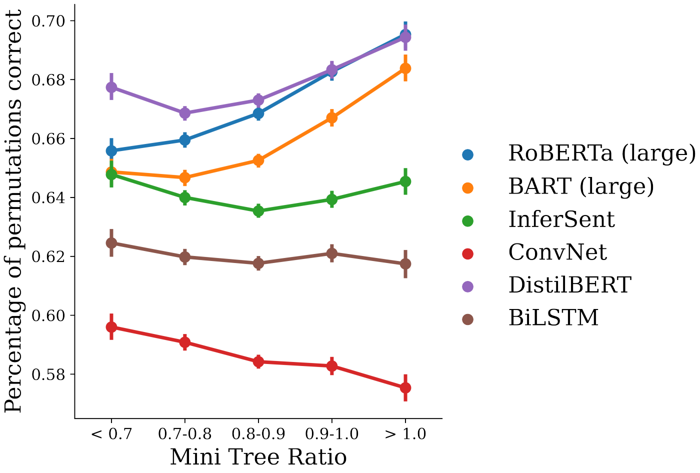


Figure 4.4 POS Tag Mini Tree overlap score and percentage of permutations which the models assigned the gold-label.

Concretely, $\beta^k_{\{w_i,S_i\}}$ computes the overlap of top-k POS tags in the neighborhood of a word w_i in S with that of the train statistic. If a word has the same mini-tree in a given sentence as it has in the training set, then the overlap would be 1. For a given sentence S_i, the aggregate $\beta^k_{\{S_i\}}$ is defined by the average of the overlap scores of all its words: $\beta^k_{\{S_i\}} = \frac{1}{|S_i|}\sum_{w_i \in S_i} \beta^k_{\{w_i,S_i\}}$, and we call it a POS minitree *signature*. We can also compute the POS minitree signature of a permuted sentence $\hat{S}_i$ to have $\beta^k_{\{\hat{S}_i\}}$. If the permuted sentence POS signature comes close to that of the true sentence, then their ratio (i.e., $\beta^k_{\{\hat{S}_i\}}/\beta^k_{\{S_i\}}$) will be close to 1. Also, since POS signature is computed with respect to the train distribution, a ratio of > 1 indicates that the permuted sentence is closer to the overall train statistic than to the original unpermuted sentence in terms of POS signature. If high overlap with the training distribution correlates with percentage of permutations deemed correct, then our models treat words as if they project syntactic minitrees.

We investigate the relationship with percentage of permuted sentences accepted with $\beta^k_{\{\hat{S}_i\}}/\beta^k_{\{S_i\}}$ in Figure 4.4. We observe that the POS Tag Minitree hypothesis holds for Transformer-based models, RoBERTa, BART and DistilBERT, where the percentage of accepted pairs increase as the sentences have higher overlap with the un-permuted sentence in terms of POS signature. For non-Transformer models such as InferSent, ConvNet, and BiLSTM models, the POS signature ratio to percentage of correct permutation remains the same or decreases, suggesting that the reasoning process employed by these models does not preserve local abstract syntax structure (i.e., POS neighbor relations).

4.5.2 Human Evaluation

We expect humans to struggle with UNLI, given our intuitions and the sentence superiority findings (but see Mollica et al. 2020). To test this, we presented two experts in NLI (one a linguist) with permuted sentence pairs to label.[4] Concretely, we draw equal number of examples from MNLI Matched dev set (100 examples where RoBERTa predicts the gold label, D^c and 100 examples where it fails to do so, D^f), and then permute these examples using $\mathcal{F}$. The experts were given no additional information (recall that it is common knowledge that NLI is a roughly balanced 3-way classification task). Unbeknownst to the experts, all permuted sentences in the sample were actually accepted by the RoBERTa (large) model (trained on MNLI dataset). We observe that the experts performed much worse than RoBERTa (Table 4.4), although their accuracy was a bit higher than random. We also find that for both experts, accuracy on permutations from D^c was higher than on D^f, which verifies findings that showed high word overlap can give hints about the ground truth label [Dasgupta et al., 2018, Poliak et al., 2018, Gururangan et al., 2018a, Naik et al., 2019].

[4] Concurrent work by Gupta et al. [2021] found that untrained crowdworkers accept NLI examples that have been subjected to different kinds of perturbations at roughly most frequent class levels—i.e., only 35% of the time.

Evaluator	Accuracy	Macro F1	Acc on D^c	Acc on D^f
X	0.581 ± 0.068	0.454	0.649 ± 0.102	0.515 ± 0.089
Y	0.378 ± 0.064	0.378	0.411 ± 0.098	0.349 ± 0.087

Table 4.4 Human (expert) evaluation on 200 permuted examples from the MNLI matched development set. Half of the permuted pairs contained shorter sentences and the other, longer ones. All permuted examples were assigned the gold label by RoBERTa-Large.

4.5.3 Training by Maximizing Entropy

We propose an initial attempt to mitigate the effect of correct prediction on permuted examples. As we observe in §4.4.2, model entropy on permuted examples is significantly lower than expected. Neural networks tend to output higher confidence than random for even unknown inputs Gandhi and Lake [2020], which might be an underlying cause of the high Permutation Acceptance.

An ideal model would be ambivalent about randomized ungrammatical sentences. Thus, we train NLI models baking in the principle of mutual exclusivity [Gandhi and Lake, 2020] by maximizing model entropy. Concretely, we fine-tune RoBERTa on MNLI while maximizing the entropy ($\mathcal{H}$) on a subset of n randomized examples ($(\hat{p}_i, \hat{r}_i)$, for each example (p, h) in MNLI. We modify the loss function as follows:

$$\mathcal{L} = \underset{\theta}{\operatorname{argmin}} \sum_{((p,h),y)} y \log(p(y|(p,h);\theta)) + \sum_{i=1}^{n} \mathcal{H}\left(y|(\hat{p}_i, \hat{h}_i);\theta\right) \tag{4.5}$$

Using this maximum entropy method ($n = 1$), we find that the model improves considerably with respect to its robustness to randomized sentences, all while taking no hit to accuracy (Table 4.5). We observe that no model reaches a Ω_{max} score close to 0, suggesting further room to explore other methods for decreasing models' Permutation Acceptance. Similar approaches have also proven useful [Gupta et al., 2021] for other tasks as well.

Eval Dataset	$\mathcal{A}$ (V)	$\mathcal{A}$ (ME)	Ω_{max} (V)	Ω_{max} (ME)
MNLI_m_dev	0.905	0.908	0.984	0.328
MNLI_mm_dev	0.901	0.903	0.985	0.329
SNLI_test	0.882	0.888	0.983	0.329
SNLI_dev	0.879	0.887	0.984	0.333
ANLI_r1_dev	0.456	0.470	0.890	0.333
ANLI_r2_dev	0.271	0.258	0.880	0.333
ANLI_r3_dev	0.268	0.243	0.892	0.334

Table 4.5 NLI Accuracy ($\mathcal{A}$) and Permutation Acceptance metrics (Ω_{max}) of RoBERTa when trained on MNLI dataset using vanilla (V) and Maximum Random Entropy (ME) method.

4.6 Related Work

Researchers in NLP have realized the importance of syntactic structure in neural networks going back to Tabor [1994]. An early hand annotation effort on PASCAL RTE [Dagan et al., 2006] suggested that "syntactic information alone was sufficient to make a judgment" for roughly one third of examples [Vanderwende and Dolan, 2005]. Anecdotally, large generative language models like GPT-2 or -3 exhibit a seemingly human-like ability to generate fluent and grammatical text [Goldberg, 2019, Wolf, 2019]. However, the jury is still out as to whether transformers genuinely acquire syntax.

Models appear to have acquired syntax. When researchers have peeked inside Transformer LM's pretrained representations, familiar syntactic structure [Hewitt and Manning, 2019, Jawahar et al., 2019, Lin et al., 2019, Warstadt and Bowman, 2020, Wu et al., 2020], or a familiar order of linguistic operations [Jawahar et al., 2019, Tenney et al., 2019], has appeared. There is also evidence, notably from agreement attraction phenomena [Linzen et al., 2016] that transformer-based models pretrained on LM do acquire some knowledge of natural language syntax [Gulordava et al., 2018, Chrupała and Alishahi, 2019, Jawahar et al., 2019, Lin et al., 2019, Manning et al., 2020, Hawkins et al., 2020, Linzen and Baroni, 2021]. Results from other phenomena [Warstadt and Bowman, 2020] such as NPI licensing [Warstadt et al., 2019a] lend additional support. The claim that LMs acquire some syntactic knowledge has been made not only for transformers, but also for convolutional neural nets [Bernardy and Lappin, 2017], and RNNs [Gulordava et al., 2018, van Schijndel and Linzen, 2018, Wilcox et al., 2018, Zhang and Bowman, 2018, Prasad et al., 2019, Ravfogel et al., 2019]—although there are many caveats (e.g., Ravfogel et al. 2018, White et al. 2018, Davis and van Schijndel 2020, Chaves 2020, Da Costa and Chaves 2020, Kodner and Gupta 2020).

Models appear to struggle with syntax. Several works have cast doubt on the extent to which NLI models in particular know syntax (although each work adopts a slightly different idea of what "knowing syntax" entails). For example, McCoy et al. [2019] argued that the knowledge acquired by models trained on NLI (for at least some popular datasets) is actually not as syntactically sophisticated as it might have initially seemed; some transformer models rely mainly on simpler, non-humanlike heuristics. In general, transformer LM performance has been found to be patchy and variable across linguistic phenomena [Dasgupta et al., 2018, Naik et al., 2018, An et al., 2019, Ravichander et al., 2019, Jeretic et al., 2020]. This is especially true for syntactic phenomena [Marvin and Linzen, 2018, Hu et al., 2020b, Gauthier et al., 2020, McCoy et al., 2020, Warstadt et al., 2020], where transformers are, for some phenomena and settings, worse than RNNs [van Schijndel et al., 2019]. From another angle, many have explored architectural approaches for increasing a network's sensitivity to syntactic structure [Chen et al., 2017, Li et al., 2020]. Williams et al. [2018a] showed that learning jointly to perform NLI and to parse resulted in parse trees that match no popular syntactic formalisms. Furthermore, models trained explicitly to differentiate acceptable sentences from unacceptable ones (i.e., one of the most common syntactic tests used by linguists) have, to date, come nowhere near human performance [Warstadt et al., 2019b].

Insensitivity to Perturbation. Most relatedly, several concurrent works [Pham et al., 2020, Alleman et al., 2021, Gupta et al., 2021, Sinha et al., 2021a, Parthasarathi et al., 2021] investigated the effect of word order permutations on transformer NNs. Pham et al. [2020] is very nearly a proper subset of our work except for investigating additional tasks (i.e. from the GLUE benchmark of Wang et al. 2018) and performing a by-layer-analysis. Gupta et al. [2021] also relies on the GLUE benchmark, but additionally investigates other types of "destructive" perturbations. Our contribution differs from these works in that we additionally include the following: we (i) outline theoretically-informed predictions for how models *should be expected* to react to permuted input (we outline a few options), (ii) show that permuting can "flip" an incorrect prediction to a correct one, (iii) show that the problem isn't specific to Transformers, (iv) show that the problem persists on out of domain data, (v) offer a suite of flexible metrics, and (vi) analyze *why* models might be accepting permutations (BLEU and POS-tag neighborhood analysis). Finally, we replicate our findings in another language. While our work (and Pham et al., Gupta et al.) only permutes data during fine-tuning and/or evaluation, recently Sinha et al. explored the sensitivity during pre-training, and found that models trained on n-gram permuted sentences perform remarkably close to regular MLM pre-training. In the context of generation, Parthasarathi et al. [2021] crafted linguistically relevant perturbations (on the basis of part-of-speech tagging and dependency parsing) to evaluate whether permutation hinders automatic machine translation models. Relatedly, but not for translation, Alleman et al. [2021] investigated a smaller inventory of perturbations with emphasis on phrasal boundaries and the effects of n-gram perturbations on different layers in the network.

NLI Models are very sensitive to words. NLI models often over-attend to particular words to predict the correct answer [Gururangan et al., 2018a, Clark et al., 2019]. Wallace et al. [2019] show that some short sequences of non-human-readable text can fool many NLU models, including NLI models trained on SNLI, into predicting a specific label. In fact, Ettinger [2020] observed that for one of three test sets, BERT loses some accuracy in word-perturbed sentences, but that there exists a subset of examples for which BERT's accuracy remains intact. If performance isn't affected (or if permutation helps, as we find it does in some cases), it suggests that these state-of-the-art models actually perform somewhat similarly to bag-of-words models Blei et al. [2003], Mikolov et al. [2013].

4.7 Discussion

4.8 Follow-up findings in the community

Chapter 5

Probing syntax understanding through distributional hypothesis

Paper: **?**

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

Anne Abeille. Lexical and syntactic rules in a Tree Adjoining Grammar. In *28th Annual Meeting of the Association for Computational Linguistics*, pages 292–298, Pittsburgh, Pennsylvania, USA, June 1990. Association for Computational Linguistics. doi: 10.3115/981823.981860. URL `https://www.aclweb.org/anthology/P90-1037`.

Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. Analyzing the behavior of visual question answering models. *arXiv preprint arXiv:1606.07356*, 2016.

Matteo Alleman, Jonathan Mamou, Miguel A Del Rio, Hanlin Tang, Yoon Kim, and SueYeon Chung. Syntactic perturbations reveal representational correlates of hierarchical phrase structure in pretrained language models. *arXiv preprint arXiv:2104.07578*, 2021. URL `https://arxiv.org/abs/2104.07578`.

Aixiu An, Peng Qian, Ethan Wilcox, and Roger Levy. Representation of constituents in neural language models: Coordination phrase as a case study. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2888–2899, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1287. URL `https://www.aclweb.org/anthology/D19-1287`.

Alan D Baddeley, Graham J Hitch, and Richard J Allen. Working memory and binding in sentence recall. *Journal of Memory and Language*, 2009. URL https://doi.org/10.1016/j.jml.2009.05.004.

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. URL https://openreview.net/forum?id=HkezXnA9YX.

Douglas K Bemis and Liina Pylkkänen. Basic linguistic composition recruits the left anterior temporal lobe and left angular gyrus during both listening and reading. *Cerebral Cortex*, 2013.

Emily M. Bender and Alexander Koller. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.463. URL https://www.aclweb.org/anthology/2020.acl-main.463.

Jean-Phillipe Bernardy and Shalom Lappin. Using deep neural networks to learn syntactic agreement. In *Linguistic Issues in Language Technology, Volume 15, 2017*. CSLI Publications, 2017. URL https://www.aclweb.org/anthology/2017.lilt-15.3.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *JMLR*, 2003. URL https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf.

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, Lisbon, Portugal, September 2015a. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL https://www.aclweb.org/anthology/D15-1075.

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, 2015b.

Joan Bresnan, Ash Asudeh, Ida Toivonen, and Stephen Wechsler. *Lexical-functional syntax*. John Wiley & Sons, 2015.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

James McKeen Cattell. The time it takes to see and name objects. *Mind*, os-XI(41):63–65, 01 1886. ISSN 0026-4423. doi: 10.1093/mind/os-XI.41.63. URL https://doi.org/10.1093/mind/os-XI.41.63.

Rui Chaves. What don't RNN language models learn about filler-gap dependencies? In *Proceedings of the Society for Computation in Linguistics 2020*, pages 1–11, New York, New York, January 2020. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2020.scil-1.1`.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1152. URL `https://www.aclweb.org/anthology/P17-1152`.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014.

Grzegorz Chrupała and Afra Alishahi. Correlating neural and symbolic representations of language. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2952–2962, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1283. URL `https://www.aclweb.org/anthology/P19-1283`.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4828. URL `https://www.aclweb.org/anthology/W19-4828`.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*, 2008. URL `https://dl.acm.org/doi/10.1145/1390156.1390177`.

Cleo Condoravdi, Dick Crouch, Valeria de Paiva, Reinhard Stolle, and Daniel G. Bobrow. Entailment, intensionality and text understanding. In *Proceedings of the HLT-NAACL 2003 Workshop on Text Meaning*, pages 38–45, 2003. URL `https://www.aclweb.org/anthology/W03-0906`.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1070. URL `https://www.aclweb.org/anthology/D17-1070`.

Jillian Da Costa and Rui Chaves. Assessing the ability of transformer-based neural models to represent structurally unbounded dependencies. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 12–21, New York, New York, January 2020. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2020.scil-1.2`.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*. Springer, 2005. URL `https://dl.acm.org/doi/10.1007/11736790_9`.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL recognising textual entailment challenge. In *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*. Springer, 2006.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=Syg-YfWCW`.

Ishita Dasgupta, Demi Guo, Andreas Stuhlmüller, Samuel J Gershman, and Noah D Goodman. Evaluating compositionality in sentence embeddings. In *Proceedings of Annual Meeting of the Cognitive Science Society*, 2018. URL `https://arxiv.org/abs/1802.04302`.

Forrest Davis and Marten van Schijndel. Recurrent neural network language models always learn English-like relative clause attachment. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1979–1990, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.179. URL `https://www.aclweb.org/anthology/2020.acl-main.179`.

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.

Allyson Ettinger. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48, 2020. doi: 10.1162/tacl_a_00298. URL `https://www.aclweb.org/anthology/2020.tacl-1.3`.

Jerry A Fodor and Zenon W Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71, 1988.

Herve Gallaire and Jack Minker. *Logic and Data Bases*. Perseus Publishing, 1978.

Kanishk Gandhi and Brenden M Lake. Mutual exclusivity as a challenge for deep neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 14182–14192. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/a378383b89e6719e15cd1aa45478627c-Paper.pdf.

Jon Gauthier, Jennifer Hu, Ethan Wilcox, Peng Qian, and Roger Levy. SyntaxGym: An online platform for targeted evaluation of language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 70–76, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-demos.10. URL https://www.aclweb.org/anthology/2020.acl-demos.10.

Yoav Goldberg. Assessing BERT's syntactic abilities. *arXiv preprint arXiv:1901.05287*, 2019. URL https://arxiv.org/abs/1901.05287.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. Colorless green recurrent networks dream hierarchically. 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 1195–1205, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1108. URL https://www.aclweb.org/anthology/N18-1108.

Ashim Gupta, Giorgi Kvernadze, and Vivek Srikumar. BERT & family eat word salad: Experiments with text understanding. *AAAI*, 2021. URL https://arxiv.org/abs/2101.03453.

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, New Orleans, Louisiana, June 2018a. Association for Computational Linguistics. doi: 10.18653/v1/N18-2017. URL https://www.aclweb.org/anthology/N18-2017.

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, 2018b.

Will Hamilton, Payal Bajaj, Marinka Zitnik, Dan Jurafsky, and Jure Leskovec. Embedding logical queries on knowledge graphs. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 2026–2037. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/7473-embedding-logical-queries-on-knowledge-graphs.pdf.

Zellig S Harris. Distributional structure. *Word*, 1954.

Robert Hawkins, Takateru Yamakoshi, Thomas Griffiths, and Adele Goldberg. Investigating representations of verb bias in neural language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4653–4663, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.376. URL https://www.aclweb.org/anthology/2020.emnlp-main.376.

Irene Heim and Angelika Kratzer. *Semantics in generative grammar*. Blackwell Oxford, 1998.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701, 2015.

John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1419. URL https://www.aclweb.org/anthology/N19-1419.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020. URL https://doi.org/10.5281/zenodo.1212303.

Hai Hu, Kyle Richardson, Liang Xu, Lu Li, Sandra Kübler, and Lawrence Moss. OCNLI: Original Chinese Natural Language Inference. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3512–3526, Online, November 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.314. URL https://www.aclweb.org/anthology/2020.findings-emnlp.314.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. A systematic assessment of syntactic generalization in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1725–1744, Online, July 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.158. URL https://www.aclweb.org/anthology/2020.acl-main.158.

Drew Arad Hudson and Christopher D. Manning. Compositional attention networks for machine reasoning. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=S1Euwz-Rb`.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1356. URL `https://www.aclweb.org/anthology/P19-1356`.

Paloma Jeretic, Alex Warstadt, Suvrat Bhooshan, and Adina Williams. Are natural language inference models IMPPRESsive? Learning IMPlicature and PRESupposition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8690–8705, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.768. URL `https://www.aclweb.org/anthology/2020.acl-main.768`.

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.

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.

Ronald M Kaplan and Joan Bresnan. Formal system for grammatical representation. *Formal Issues in Lexical-Functional Grammar*, 1995.

Dimitri Kartsaklis, Mohammad Taher Pilehvar, and Nigel Collier. Mapping text to knowledge graph entities using multi-sense lstms. *arXiv preprint arXiv:1808.07724*, 2018.

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.

Yigitcan Kaya, Sanghyun Hong, and Tudor Dumitras. Shallow-Deep Networks: Understanding and mitigating network overthinking. In *Proceedings of the 2019 International Conference on Machine Learning (ICML)*, Long Beach, CA, Jun 2019. URL https://arxiv.org/abs/1810.07052.

Jordan Kodner and Nitish Gupta. Overestimation of syntactic representation in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1757–1762, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.160. URL https://www.aclweb.org/anthology/2020.acl-main.160.

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.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://www.aclweb.org/anthology/2020.acl-main.703.

Peiguang Li, Hongfeng Yu, Wenkai Zhang, Guangluan Xu, and Xian Sun. SA-NLI: A supervised attention based framework for natural language inference. *Neurocomputing*, 2020. URL `https://doi.org/10.1016/j.neucom.2020.03.092`.

Yongjie Lin, Yi Chern Tan, and Robert Frank. Open sesame: Getting inside BERT's linguistic knowledge. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 241–253, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4825. URL `https://www.aclweb.org/anthology/W19-4825`.

Tal Linzen and Marco Baroni. Syntactic structure from deep learning. *Annual Review of Linguistics*, 2021. URL `https://doi.org/10.1146/annurev-linguistics-032020-051035`.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535, 2016. doi: 10.1162/tacl_a_00115. URL `https://www.aclweb.org/anthology/Q16-1037`.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *arXiv*, 2019. URL `https://arxiv.org/abs/1907.11692`.

Christopher D. Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. Emergent linguistic structure in artificial neural networks trained by self-supervision. *Proceedings of the National Academy of Sciences*, 117(48):30046–30054, 2020. ISSN 0027-8424. doi: 10.1073/pnas.1907367117. URL `https://www.pnas.org/content/117/48/30046`.

Rebecca Marvin and Tal Linzen. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language*

Processing, pages 1192–1202, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1151. URL https://www.aclweb.org/anthology/D18-1151.

R. Thomas McCoy, Junghyun Min, and Tal Linzen. BERTs of a feather do not generalize together: Large variability in generalization across models with similar test set performance. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 217–227, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.blackboxnlp-1.21. URL https://www.aclweb.org/anthology/2020.blackboxnlp-1.21.

Tom McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1334. URL https://www.aclweb.org/anthology/P19-1334.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013. URL http://arxiv.org/abs/1301.3781.

Alexander Miller, Will Feng, Dhruv Batra, Antoine Bordes, Adam Fisch, Jiasen Lu, Devi Parikh, and Jason Weston. Parlai: A dialog research software platform. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 79–84, 2017.

Francis Mollica, Matthew Siegelman, Evgeniia Diachek, Steven T Piantadosi, Zachary Mineroff, Richard Futrell, Hope Kean, Peng Qian, and Evelina Fedorenko. Composition is the core driver of the language-selective network. *Neurobiology of Language*, 1(1):104–134, 2020. URL https://direct.mit.edu/nol/article/1/1/104/10024/Composition-is-the-Core-Driver-of-the-Language.

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.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C18-1198.

Aakanksha Naik, Abhilasha Ravichander, Carolyn Rose, and Eduard Hovy. Exploring numeracy in word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3374–3380, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1329. URL https://www.aclweb.org/anthology/P19-1329.

Nikita Nangia and Samuel R. Bowman. Human vs. muppet: A conservative estimate of human performance on the GLUE benchmark. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4566–4575, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1449. URL https://www.aclweb.org/anthology/P19-1449.

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.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.441. URL `https://www.aclweb.org/anthology/2020.acl-main.441`.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-4009. URL `https://www.aclweb.org/anthology/N19-4009`.

Prasanna Parthasarathi, Koustuv Sinha, Joelle Pineau, and Adina Williams. Sometimes we want translationese. *arXiv preprint arXiv:2104.07623*, 2021. URL `https://arxiv.org/abs/2104.07623`.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

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

Thang M Pham, Trung Bui, Long Mai, and Anh Nguyen. Out of order: How important is the sequential order of words in a sentence in natural language understanding tasks? *arXiv preprint arXiv:2012.15180*, 2020. URL `https://arxiv.org/abs/2012.15180`.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/S18-2023. URL `https://www.aclweb.org/anthology/S18-2023`.

Carl Pollard and Ivan A Sag. *Head-driven phrase structure grammar*. University of Chicago Press, 1994. URL `https://web.stanford.edu/group/cslipublications/cslipublications/site/0226674479.shtml`.

Grusha Prasad, Marten van Schijndel, and Tal Linzen. Using priming to uncover the organization of syntactic representations in neural language models. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 66–76, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/K19-1007. URL `https://www.aclweb.org/anthology/K19-1007`.

Liina Pylkkänen, Douglas K Bemis, and Estibaliz Blanco Elorrieta. Building phrases in language production: An meg study of simple composition. *Cognition*, 2014.

J R Quinlan. Learning logical definitions from relations. *Mach. Learn.*, 5(3):239–266, August 1990. ISSN 0885-6125. doi: 10.1007/BF00117105.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 2019. URL `https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf`.

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.

Shauli Ravfogel, Yoav Goldberg, and Francis Tyers. Can LSTM learn to capture agreement? the case of Basque. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 98–107, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5412. URL `https://www.aclweb.org/anthology/W18-5412`.

Shauli Ravfogel, Yoav Goldberg, and Tal Linzen. Studying the inductive biases of RNNs with synthetic variations of natural languages. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3532–3542, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1356. URL `https://www.aclweb.org/anthology/N19-1356`.

Abhilasha Ravichander, Aakanksha Naik, Carolyn Rose, and Eduard Hovy. EQUATE: A benchmark evaluation framework for quantitative reasoning in natural language inference. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 349–361, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/K19-1033. URL `https://www.aclweb.org/anthology/K19-1033`.

Matthew Richardson, Christopher JC Burges, and Erin Renshaw. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, 2013.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 2020. doi: 10.1162/tacl_a_00349. URL `https://www.aclweb.org/anthology/2020.tacl-1.54`.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020. URL `https://arxiv.org/abs/1910.01108`.

Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In *Advances in neural information processing systems*, pages 4967–4976, 2017.

Adam Santoro, Ryan Faulkner, David Raposo, Jack Rae, Mike Chrzanowski, Theophane Weber, Daan Wierstra, Oriol Vinyals, Razvan Pascanu, and Timothy Lillicrap. Relational recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 7310–7321, 2018.

Yves Schabes, Anne Abeille, and Aravind K. Joshi. Parsing strategies with 'lexicalized' grammars: Application to Tree Adjoining Grammars. In *Coling Budapest 1988 Volume 2: International Conference on Computational Linguistics*, 1988. URL `https://www.aclweb.org/anthology/C88-2121`.

Eckart Scheerer. Early german approaches to experimental reading research: The contributions of wilhelm wundt and ernst meumann. *Psychological Research*, 1981.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. CLUTRR: A diagnostic benchmark for inductive reasoning from text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4506–4515, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1458. URL `https://aclanthology.org/D19-1458`.

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. *arXiv preprint arXiv:2104.06644*, 2021a. URL `https://arxiv.org/abs/2104.06644`.

Koustuv Sinha, Prasanna Parthasarathi, Joelle Pineau, and Adina Williams. UnNatural Language Inference. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7329–7346, Online, August 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.569. URL `https://aclanthology.org/2021.acl-long.569`.

Joshua Snell and Jonathan Grainger. The sentence superiority effect revisited. *Cognition*, 2017.

Joshua Snell and Jonathan Grainger. Word position coding in reading is noisy. *Psychonomic bulletin & review*, 26(2):609–615, 2019.

Shagun Sodhani, Sarath Chandar, and Yoshua. Bengio. On Training Recurrent Neural Networks for Lifelong Learning. *arXiv e-prints*, November 2018.

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.

Whitney Tabor. *Syntactic innovation: A connectionist model.* PhD thesis, 1994. URL https://www.proquest.com/openview/0a8f7e8a71e058b12053b545ca857fb2/1.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL https://www.aclweb.org/anthology/P19-1452.

Hiroshi Toyota. Changes in the constraints of semantic and syntactic congruity on memory across three age groups. *Perceptual and Motor Skills*, 2001. URL https://pubmed.ncbi.nlm.nih.gov/11453195/.

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. doi: 10.1016/B978-0-12-800077-9/00020-7.

Marten van Schijndel and Tal Linzen. A neural model of adaptation in reading. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4704–4710, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1499. URL https://www.aclweb.org/anthology/D18-1499.

Marten van Schijndel, Aaron Mueller, and Tal Linzen. Quantity doesn't buy quality syntax with neural language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5831–5837, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1592. URL https://www.aclweb.org/anthology/D19-1592.

Lucy Vanderwende and William B Dolan. What syntax can contribute in the entailment task. In *Machine Learning Challenges Workshop*. Springer, 2005. URL https://dl.acm.org/doi/10.1007/11736790_11.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal adversarial triggers for attacking and analyzing NLP. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2153–2162, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1221. URL https://www.aclweb.org/anthology/D19-1221.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL https://www.aclweb.org/anthology/W18-5446.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *NeurIPS*, 2019. URL `https://papers.nips.cc/paper/2019/hash/4496bf24afe7fab6f046bf4923da8de6-Abstract.html`.

Z. Wang, L. Li, D. D. Zeng, and Y. Chen. Attention-based multi-hop reasoning for knowledge graph. In *2018 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 211–213, Nov 2018. doi: 10.1109/ISI.2018.8587330.

Alex Warstadt and Samuel R Bowman. Can neural networks acquire a structural bias from raw linguistic data? In *Proceedings of the 42nd Annual Virtual Meeting of the Cognitive Science Society*, 2020. URL `https://arxiv.org/abs/2007.06761`.

Alex Warstadt, Yu Cao, Ioana Grosu, Wei Peng, Hagen Blix, Yining Nie, Anna Alsop, Shikha Bordia, Haokun Liu, Alicia Parrish, Sheng-Fu Wang, Jason Phang, Anhad Mohananey, Phu Mon Htut, Paloma Jeretic, and Samuel R. Bowman. Investigating BERT's knowledge of language: Five analysis methods with NPIs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2877–2887, Hong Kong, China, November 2019a. Association for Computational Linguistics. doi: 10.18653/v1/D19-1286. URL `https://www.aclweb.org/anthology/D19-1286`.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, March 2019b. doi: 10.1162/tacl_a_00290. URL `https://www.aclweb.org/anthology/Q19-1040`.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392, 2020. doi: 10.1162/tacl_a_00321. URL `https://www.aclweb.org/anthology/2020.tacl-1.25`.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association of Computational Linguistics*, 6:287–302, 2018.

Yun Wen, Joshua Snell, and Jonathan Grainger. Parallel, cascaded, interactive processing of words during sentence reading. *Cognition*, 2019.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. Towards AI-Complete question answering: A set of prerequisite toy tasks. 2015. ISSN 0378-7753. doi: 10.1016/j.jpowsour.2014.09.131.

Aaron Steven White, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. Lexicosyntactic inference in neural models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4717–4724, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1501. URL `https://www.aclweb.org/anthology/D18-1501`.

Ethan Wilcox, Roger Levy, Takashi Morita, and Richard Futrell. What do RNN language models learn about filler–gap dependencies? In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 211–221, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5423. URL `https://www.aclweb.org/anthology/W18-5423`.

Adina Williams, Andrew Drozdov, and Samuel R. Bowman. Do latent tree learning models identify meaningful structure in sentences? *Transactions of the Association for Computational Linguistics*, 6:253–267, 2018a. doi: 10.1162/tacl_a_00019. URL `https://www.aclweb.org/anthology/Q18-1019`.

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 2018b. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL `https://www.aclweb.org/anthology/N18-1101`.

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, 2018c.

Thomas Wolf. Some additional experiments extending the tech report" assessing berts syntactic abilities" by yoav goldberg. Technical report, HuggingFace, 2019. URL `https://huggingface.co/bert-syntax/extending-bert-syntax.pdf`.

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. Transformers: State-of-the-art natural language processing. In *EMNLP: System Demonstrations*, 2020.

Zhiyong Wu, Yun Chen, Ben Kao, and Qun Liu. Perturbed masking: Parameter-free probing for analyzing and interpreting BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4166–4176, Online, July 2020.

Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.383. URL https://www.aclweb.org/anthology/2020.acl-main.383.

Wenhan Xiong, Thien Hoang, and William Yang Wang. Deeppath: A reinforcement learning method for knowledge graph reasoning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 564–573, 2017.

Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. One-shot relational learning for knowledge graphs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1980–1990, 2018.

Kelly Zhang and Samuel Bowman. Language modeling teaches you more than translation does: Lessons learned through auxiliary syntactic task analysis. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 359–361, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5448. URL https://www.aclweb.org/anthology/W18-5448.

Han Zhao, Zhengdong Lu, and Pascal Poupart. Self-adaptive hierarchical sentence model. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 4069–4076, 2015. URL https://arxiv.org/abs/1504.05070.

Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian McAuley, Ke Xu, and Furu Wei. Bert loses patience: Fast and robust inference with early exit. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18330–18341. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/d4dd111a4fd973394238aca5c05bebe3-Paper.pdf.

Glossary

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

Acronyms

LLMs Large Language Models. 2

NLU Natural Language Understanding. 4, 5, 24, 32–34

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) :par

(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=non
```

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`!