Investigating Syntactic Enhancements in LLMs with Graph Convolutional Networks for Natural Language Inference

Master Artificial Intelligence, Utrecht University

Author: Luca Lin, 5522146
Daily Supervisor: Dr. Lasha Abzianidze
Second Examiner: Dr. Denis Paperno

July 14, 2023

Utrecht University

Contents

Abstract

This Master's Thesis presents an exploration of incorporating syntax trees into pre-trained Large Language Models (LLMs) for the task of Natural Language Inference (NLI). NLI is an important task for evaluating language models' ability to predict the entailment relationship between two sentences, thus showcasing a model's capacity for Natural Language Understanding (NLU). This study predominantly focuses on the BERT-base-uncased model, assessing the effects of enhancing it with an inductive bias toward linguistically derived syntactic trees using Graph Convolutional Networks, and the effects on performance on various NLI benchmark datasets and out-of-domain evaluation sets. While earlier research has delved into the impacts of enhancing LLMs with dependency structures, the effects of incorporating constituency structures and combining both parsing techniques remain largely unexplored. Experimental results reveal that while enhancement of BERT with syntactic structures does not notably benefit generic large-scale NLI datasets, it significantly aids models in scenarios where the underlying syntactic structure is important for the inference task, such as in semi-automatically generated datasets. This is particularly evident when training data is scarce, a common challenge in many real-world applications. Results further show that of the two investigated syntactic structures, constituency structures provide the most benefits in learning representations for monotonicity reasoning, an important skill that requires the ability to capture interactions between lexical and syntactic structures. Furthermore, we demonstrate that constituency parsing can help the BERT model learn useful representations for the syntactic structure of passive sentences, an area identified in previous research as a shortcoming of BERT.

1 Introduction

Artificial Intelligence (AI) is an interdisciplinary field with the goal of understanding the nature of intelligence and replicating it in computer systems. Advancements in AI have become increasingly important for many aspects of society, from impacting the way our education system works (Zhai et al., 2021) to increasing productivity in professions such as software programming (Peng et al., 2023). Since its foundation at The Dartmouth Summer Research Project on AI in 1956[1], the field of AI has made tremendous progress and has split into various subfields, such as Computer Vision and Natural Language Processing (NLP), each tackling and researching various aspects of biological and human intelligence. Specifically, NLP, operating at the intersection between linguistics and computer science, grants machines the ability to perform a wide variety of language-related tasks, such as Language Translation, Fact Checking, Question Answering (QA), and Language Generation. As such, NLP emerges as a critical subfield of AI, attempting to simulate a unique aspect of human intelligence within computational systems: understanding and using natural language.

[1] Darthmouth Workshop

Understanding and reasoning with language are central aspects of human intelligence. However, imbuing machines with Natural Language Understanding (NLU) poses a challenge to the field of NLP due to the ambiguity and diversity with which meaning can be expressed. Nevertheless, a system that can perform the same types of reasoning as humans can benefit many NLP tasks. More specifically, if a system can determine the semantic meaning of a proposition, and infer the truth value of another text fragment given the proposition, stands to benefit a broad spectrum of NLP applications. For instance, in QA, validating the truth value of an answer in relation to a source text could bolster the system's reliability and provide more accurate answers. For Fact Verification and Fake News Detection, validating the truth value of a statement given a trustworthy source text stands at the core of these tasks. Yet, although the various NLP tasks share this common ground, researchers working on different NLP applications have worked independently on inference processes, leading to highly specialized methodologies and a lack of overview. To address this gap, a generic, unified task was proposed by Dagan and Glickman (2004) to evaluate the NLU capabilities of language systems, termed Recognizing Textual Entailment (RTE), which subsequently evolved into Natural Language Inference (NLI).

In the original RTE setting, given a premise P and a hypothesis H, the relationship between P and H is *entailing* if a human reading P can infer that H is most likely true, and *non-entailing* if a human can not infer that H is true (Dagan and Glickman, 2004). The task has been subsequently expanded upon to include the contradiction label, indicating that H contradicts P, resulting in a ternary classification task with the labels *entailment*, *contradiction*, and *neutral*[2]. Thus, NLI seeks to assess the reasoning capabilities of language models with a generic evaluation task.

In addition to benefitting many other NLP tasks, the general-purpose objective of NLI has been found to be particularly beneficial for narrow domains with few manually annotated data. Research by Laurer et al. (2022) shows that deep transfer learning from the NLI domain to more specific domains like political sciences can reduce the data requirements for language models up to tenfold while maintaining performance levels. Therefore, research in NLI is not only directly beneficial for numerous NLP applications, but its foundational setup also significantly aids in other domains where annotated data might be scarce, thereby highlighting the pivotal role of ongoing research and innovation in NLI for the broader landscape of NLP and AI.

Besides the many benefits provided by the task of NLI, the field of NLP has seen significant advancements since NLI's inception. While traditional approaches to NLI have involved rule-based methods, the development of large-scale datasets like SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018) have allowed neural-based approaches to become competitive in the task of NLI. Moreover, the development of pre-trained Large Language Models (LLMs) such as BERT (Devlin et al., 2018)

[2]Some datasets maintain the 2-label classification because the distinction between contradiction and neutral is not sufficiently clear for the particular linguistic phenomena the dataset targets.

have greatly contributed to increased performance across many NLP tasks, including NLI. The improved performances on NLI benchmark datasets have suggested that LLM-based models can understand and reason with natural language.

Concurrent with the development of LLMs, researchers have incorporated linguistically derived syntactic structures into language models, further bolstering performance across various NLP benchmarks. Two such structures involve constituency (Chomsky, 1957) and dependency trees (Mel'cuk et al., 1988), and various methodologies have been proposed to incorporate them into language models. A popular approach is to model the tree structures using Graph Convolutional Networks (Kipf and Welling, 2016), which enable direct modeling of the graph and tree structures using a neural network-based approach, presenting an efficient way to incorporate the linguistic structures into language models. Furthermore, by combining linguistically informed features with neural networks, it is possible to reduce the number of parameters and data requirements while providing robust performance. For instance, hybrid systems such as NeuralLog (Chen et al., 2021) capitalize on the advantages of both neural networks and expert linguistic knowledge to enhance performance, robustness, and interpretability for the task of NLI. Hence, the integration of prior linguistic knowledge into neural networks can offer substantial advantages.

In spite of the advancements and increased performances on various NLI benchmarks, a growing body of literature has emerged pointing out various shortcomings of language models on NLI datasets. For example, research by McCoy et al. (2019) shows that language models adopt fallible and shallow heuristics, and they create the Heuristic Analysis for NLI Systems (HANS) dataset to make the evaluation of the use of such fallible heuristics more accessible. Likewise, Yanaka et al. (2019a) show that language models struggle with modeling the linguistic phenomena of monotonicity reasoning, creating the monotonicity-driven datasets of HELP and Monotonicity Entailment Dataset (MED) in the process. Therefore, the research on the shortcomings of language models on NLI datasets has brought to question whether language models can truly understand and reason with natural language. Moreover, the evaluation of language models enhanced with linguistic structures has been limited to large-scale, generic benchmark datasets, leaving open questions as to whether they are still relying on shallow heuristics, and whether the linguistic structures help LLM-based models understand monotonicity reasoning.

This thesis investigates the effects of enhancing LLMs with linguistically derived constituency and dependency structures via GCNs and evaluates the effects of doing so for the task of NLI. We train models enhanced with dependency structures, constituency structures, and a combination of both structures and evaluate their performances on various generic NLI benchmark datasets, as well as the evaluation sets of HANS and MED designed to test the model's ability to handle specific linguistic phenomena. Through a series of in-domain, out-of-domain, and transfer learning experiments, this study aims at illuminating the effects of enhancing LLMs with syntactic structures on their reasoning and inference capabilities for the task of

NLI under different dataset distributions, thereby showing the potential usefulness of enhancing LLMs with linguistically informed structures for the broader landscape of NLP and AI. To facilitate follow-up research and reproducibility, we make our code publicly available at https://github.com/lucalin17081994/Syntax-Enhanced-Bert.

1.1 Research Questions

Although enhancing language models with syntactic structures has been explored in the literature, several key issues remain unresolved. This thesis aims to address the following research questions and sub-questions:

- **RQ1.** What is the effect of enhancing BERT with constituency or dependency structures with GCNs on their performance for the task of NLI?

  Enhancing LLMs with constituency structures only has not been sufficiently studied for the task of NLI, and we aim to address this gap in this thesis.

  - **RQ1.a** Is there a beneficial effect when integrating both constituency and dependency structures for the task of NLI?

    Existing research has delved into the integration of both syntactic structures in the field of NLI using various methods (Bai et al., 2021; Zhou et al., 2020). Despite these efforts, the methodology involving the incorporation of both linguistic structures via GCNs remains notably unexplored. Additionally, an exhaustive evaluation of the linguistic capabilities of these models using out-of-domain test sets is still a largely untouched area of research.

  - **RQ1.b** What is the effect of fine vs coarse granularity level of the dependency labels on performance for the task of NLI?

    Previous studies have omitted the incorporation of dependency labels due to the risk of overfitting and overparameterization (Marcheggiani and Titov, 2017; He et al., 2020). However, Fei et al. (2021) show that including dependency labels improves performance on the Semantic Role Labeling task. Consequently, we experiment with different levels of label granularities to investigate whether clustering similar dependency labels can improve performance for the task of NLI.

- **RQ2.** Can enhancing BERT with dependency or constituency structures through GCNs help BERT generalize towards the HANS dataset?

  Previous research has demonstrated that dependency structures can decrease BERT's reliance on shallow heuristics on the HANS dataset He et al. (2020). Nonetheless, the effects of enhancing BERT with constituency structures or both structures on the HANS dataset remain unexplored.

- **RQ3.** Does enhancing BERT with constituency or dependency structures with GCNs help with monotonicity reasoning in the MED dataset?

  Work by Chen (2021) has shown that dependency structures can help tree-LSTMs improve performance over BERT-based models on the monotonicity reasoning dataset of MED. Nevertheless, to the best of our knowledge, the effects of enhancing BERT-based models with linguistic structures for the monotonicity reasoning task of MED have not been studied.

  - **RQ3.a** Can syntax help BERT increase performance over the baseline when trained on the monotonicity problems from HELP?

    Generic NLI datasets such as SNLI and MNLI have been found to lack monotonicity reasoning signals. Thus, numerous studies have employed the HELP dataset as a means to explore the capacity of language models to comprehend monotonicity reasoning when provided with an adequate number of examples (Yanaka et al., 2019a; Chen, 2021; Rozanova et al., 2022). Similarly, our study employs the HELP dataset to examine the potential role of linguistic structures in enhancing performance on the monotonicity reasoning task.

  - **RQ3.b** How effective are syntax-enhanced models at identifying the scope and argument structures of conjunction and disjunction operators in the MED dataset?

    The MED dataset is annotated with conjunction and disjunction instances, pointing towards the utilization of these operators for the monotonicity reasoning task. We hypothesize that linguistic structures could help our models learn the scope and argument structures of these operators, thereby enhancing performance on these cases in the MED dataset. Given that the conjunction and disjunction operators have two arguments, we further hypothesize that constituency structures may offer superior performance over dependency structures in learning these operators' scope and argument structures because they share parent nodes in the tree structure.

- **RQ4.a** What are the effects of enhancing BERT with syntax when transfer-learning from SNLI to the SICK dataset?

  The SICK dataset was created as a benchmark dataset prior to SNLI for the task of RTE to facilitate the development of Distributional Semantics Models (Marelli et al., 2014). Distributional Semantics Models make use of vectors derived from corpora using co-occurrence statistics to represent the meaning of words (Evert, 2010). SICK comprises semi-automatically generated sentences rich in lexical and syntactic information. We hypothesize that this feature makes the dataset especially valuable for models enhanced with syn-

tactic structures, as they could potentially harness the abundant syntactic information in SICK to outperform the baseline model.

- **RQ4.b** Can syntax help BERT adapt to a new distribution in a transfer learning and few-shot-learning setting?

  Laurer et al. (2022) have shown that transfer learning from the NLI domain to a more narrow domain where data is scarce can reduce the data requirements up to tenfold while maintaining performance levels. Consequently, an interesting question to explore is whether linguistic structures can provide the same benefits when transfer learning from a large-scale NLI dataset to another NLI dataset where the size of the training set is limited.

1.2 Contributions

Through our proposed research questions, our contributions to the fields of NLP and AI are as follows:

- We provide empirical data for the effects of enhancing the BERT-base-uncased LLM with syntactic features via GCNs on various NLI benchmarks and evaluation sets.
- We show that enhancing BERT with constituency, dependency, or a combination of both syntactic structures via GCNs does not benefit popular NLI benchmark datasets, but that curating the datasets of hypothesis-only biases can increase the benefits of enhancing BERT with syntactic structures.
- For datasets where the underlying syntactic structure is important for the inference task, we show that BERT enhanced with syntax via GCNs can better leverage the syntactic information and adapt to new dataset distributions over the baseline, even in settings where data is scarce.
- We show that constituency structures can help BERT learn useful representations for the linguistic phenomena of monotonicity reasoning and for passive sentence constructions, whereas dependency structures do not provide the same benefits.

1.3 Outline

In this section, we have introduced the task of NLI and its importance in the broader landscape of NLP and AI. We have highlighted the efforts of the NLP community to enhance language models with linguistically derived features, and have introduced the literature on the limitations of language models on the task of NLI. This has led to the identification of a series of research questions that we will investigate further in this thesis. In Section 2, we will delve into the details of transformer-based LLM models, the challenges and shortcomings of language models for the

task of NLI found in existing research, and the different methods of incorporating syntactic structures into language models. In Section 3, we provide details on the chosen architecture and pipeline, further clarify implementation details, and provide information on the NLI datasets used to answer our research questions. In Section 4, we detail our experimental methodologies designed to investigate the specified research questions. Section 5 provides a comprehensive report and discussion of our empirical findings, which we subsequently compare with relevant literature under similar conditions wherever possible. Section 6 discusses potential limitations encountered in our research, while Section 7 concludes this thesis by summarizing our key findings and discussing potential future research.

2 Related Work

2.1 Large Language Models

Based on the Transformer architecture (Vaswani et al., 2017), LLMs have pushed the state-of-the-art (SOTA) performance across numerous NLP benchmarks, and their versatility has since been leveraged in other fields such as Computer Vision (Dosovitskiy et al., 2020; Arnab et al., 2021), Chemistry (Wu et al., 2023), and Speech Processing (Dong et al., 2018). The Transformer's key features, such as the self-attention mechanism and contextualized embeddings, made a significant improvement over its predecessor, the LSTM (Hochreiter and Schmidhuber, 1997). Unlike the LSTM, which processes inputs sequentially and uses static embeddings like Word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014), the Transformer allows the parallel processing of the inputs. These capabilities enable self-supervised pre-training on vast amounts of data, which reduces the need for training data in specialized domains, and allows for more informed initialization of the weights, thus enhancing the model's generalization performance by bringing the weights closer to a more informed solution (Erhan et al., 2009, 2010; Balestriero et al., 2023). Moreover, the Transformer's encoder and decoder units can be hierarchically stacked and scaled, as demonstrated in families of encoder-type models like BERT (Devlin et al., 2018), and decoder-type models such as GPT (Brown et al., 2020). This hierarchical scaling enables the model to extract more abstract features from the input data, further bolstering its performance across a wide variety of tasks.

For the task of NLI, which involves the classification of sentence pairs, initial work on LSTMs encoded the premise and hypothesis separately and aligned them after the initial encoding (Chen et al., 2017). However, Devlin et al. (2018) have shown that concatenating premise and hypothesis pairs into a single input to BERT is more beneficial, as the model is able to align words between the two sentences using the self-attention mechanism and capture relationships between words by requiring minimal changes to the NLP pipeline. Many variations have been developed to

attempt to increase their performances such as RoBERTa (Liu et al., 2019), which provides more robust pre-training, OpenAI's GPT-4 (OpenAI, 2023), which scales the model as well as provides more pre-training data, and DistilBERT (Sanh et al., 2019), which reduce the training time and the number of parameters in BERT while providing performance levels similar to the original implementation. In this thesis, we use the "BERT-base-uncased" as our baseline model and as the backbone for our syntax-enhanced LLM, as BERT-base encompasses the primary mechanisms of pre-trained LLMs that are pertinent to our research.

2.2 Challenges and Shortcomings of Language Models and Natural Language Inference Datasets

With the increasing applications and performance of LLMs across many NLP tasks, a growing body of literature has emerged, pointing out various shortcomings in both the models and the NLI datasets used for training and evaluation. Works by Gururangan et al. (2018); Poliak et al. (2018) demonstrate that simple LSTM models such as InferSent (Conneau et al., 2017) and simple bag-of-words bi-gram models such as fastText (Joulin et al., 2016) are able to achieve 69% and 53.9% accuracy, well above random chance, on both validation and test sets of SNLI and MNLI respectively when only provided access to the hypothesis. In doing so, Poliak et al. (2018) propose a hypothesis-only-baseline for evaluating models on SNLI and MNLI, and Gururangan et al. (2018) introduce the term *"annotation artifacts"* to describe biases associated with the data generation process. Geva et al. (2019) observe an increased gain in performance on the MNLI dataset when the model is provided access to the annotator ID, indicating that information about the annotator writing style may leak from the training data into the evaluation data when the dataset is not carefully curated.

In light of research on annotation artifacts, McCoy et al. (2019) demonstrate that language models employ shallow heuristics based on lexical and syntactic overlaps. They develop the **Heuristic Analysis for NLI Systems** (HANS) dataset and show that language models perform below 10% accuracy in cases where applying these heuristics would make the wrong prediction. Consequently, research by Sinha et al. (2021) and Pham et al. (2021) reveal that LLM-based models are largely insensitive to word order and sequence permutations when trained on NLI datasets.

Determiners	First argument	Second argument
every, each, all	downward	upward
some, a, a few, many, several, proper noun	upward	upward
any, no, few, at most X, fewer than X, less than X	downward	downward
the, both, most, this, that	non-monotone	upward
exactly	non-monotone	non-monotone

Table 1: Examples of determiners and their polarities, Yanaka et al. (2019a).

Category	**Examples**
Determiners	*every, all, any, few, no*
Negation	*not, n't, never*
Verbs	*deny, prohibit, avoid*
Nouns	*absence of, lack of, prohibition*
Adverbs	*scarcely, hardly, rarely, seldom*
Prepositions	*without, except, but*
Conditionals	*if, when, in case that, provided that, unless*

Table 2: Examples of downward monotone operators provided by Yanaka et al. (2019a). Identifying monotonicity context requires the model to identify the monotonicity operator and the polarity of its arguments based on the syntactic structure of the sentences.

Likewise, Yanaka et al. (2019a) develop the **Monotonicity Entailment Dataset** (MED), and demonstrate that neural networks struggle to understand the linguistic phenomena of monotonicity reasoning. Monotonicity reasoning replaces constituents in the premise with either a more general concept or a more specific one to generate new hypotheses. In order for the generated hypothesis to be entailed, the replaced constituents must match the polarity (positive ↑, negative ↓, or neutral 0) of the argument position of the monotonicity operators they are under the scope of. A context is upward entailing (↑) if the monotonicity operator allows for the constituent replacement to be a more general concept. In contrast, a context is downward entailing (↓) if the monotonicity operator allows for the constituent replacement to be a more specific concept. However, if the constituent replacement violates the polarity of the monotonicity operator, the resulting sentence is non-entailing. In order to make this clear, we make use of an example provided by Yanaka et al. (2019a):

P: Every [NP person ↓] [VP bought a movie ticket ↑] (1)

H: Every young person bought a ticket (2)

In the above examples, the determiner *"Every"* is a binary monotone operator which is downward entailing in the first argument and upward entailing in the

second. Violating the properties of the monotonicity operator, for example, by replacing *"person"* in (1) with a more general concept such as *"entity"*, will lead to a non-entailed sentence. Furthermore, encapsulating the sentence with an additional downward-monotone operator such as *"When"* as in "When every person bought a movie ticket", reverses the direction of the arguments of *"Every"* to upward in the first argument, and downward in the second argument. Therefore, to be able to perform monotonicity reasoning, the model should be able to capture the interaction between lexical and syntactic structures, and further identify the monotone operators and the polarity of their arguments based on the syntactic structure of the sentence. Table 1 illustrates determiners and the polarities of their arguments, and Table 2 showcases examples of downward monotone operators. In this thesis, we investigate the effects of enhancing LLM-based models with syntax on the HANS and MED datasets, as we hypothesize that incorporating syntactic structures into LLM-based models can improve their performance on the heuristic evaluation set of HANS, and the monotonicity-driven evaluation set of MED.

2.3 Syntax-Enhanced Large Language Models

Syntactic structures, such as dependency and constituency trees, play a crucial role in understanding the grammatical structure and relationships between words in a sentence. Constituency trees decompose the sentence into its constituents and organize them into hierarchical graphs, maintaining the local and global grammatical structure of the sentence (Chomsky, 1957). Dependency structures, on the other hand, represent grammatical dependencies between two words, the head and the dependent (Mel'cuk et al., 1988). In doing so, dependency structures identify functional grammatical relations between words. An example of both types of syntactic trees can be found in Figure 1. The two types of syntactic structures view the sentence from different perspectives and have been argued in the literature to complement each other when integrated together (Fei et al., 2021).

Diverse methods have been proposed to incorporate the syntactic structures into Neural Networks, resulting in increased benchmark performances on various NLP tasks. For the task of NLI, work by Bai et al. (2021) masks the self-attention heads of BERT using structures found in both dependency and constituency trees. Zhou et al. (2020) incorporate both types of syntactic structures by means of additional pre-training and multi-task learning. Glavaš and Vulić (2021) perform intermediate pre-training (IPT) to repurpose BERT's weights to become a SOTA dependency parser, and subsequently fine-tune the model on the task of NLI. These various methodologies attempt to incorporate syntactic structures into the transformer by targeting different key features within the model's architecture.

A popular methodology for incorporating syntactic structures into language models involves directly modeling the graph structures of syntax trees via GCNs. For example, Xu et al. (2022) calculate fine-grained association graphs with dependency structures and use Neural Quadratic Assignment Programming to extract syntactic

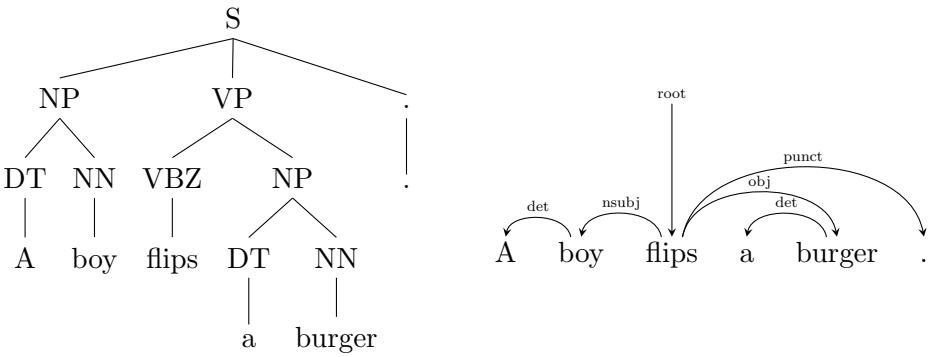

Figure 1: Example of constituency (left) and dependency (right) trees. Constituency trees represent the local and global grammatical structures in a hierarchical fashion, whereas dependency trees focus on the functional dependency relations between words.

matching patterns which are modeled using GCNs. He et al. (2020) use GCNs to combine BERT embeddings with dependency structures. For the task of Semantic Role Labeling, Fei et al. (2021) integrate both dependency and constituency structures using GCNs, and argue that the two types of structures are mutually beneficial for the task. In a survey on GCNs for various NLP tasks, Wu et al. (2021) identify 31 works in the literature incorporating dependency structures via GCNs, whereas only three works have been listed incorporating constituency structures, none of which for the task of NLI. Thus, the effects of enhancing LLM-based models with constituency trees only, and incorporating both types of syntactic structures have not been sufficiently studied, and require further exploration. In this thesis, we make use of the GCN-based implementation provided by Fei et al. (2021)[3], as the incorporation of both syntactic structures provides us with a unique opportunity to explore the effects of enhancing BERT with both types of syntactic structures for the task of NLI.

As we have chosen a neural-based approach with GCNs to incorporate syntactic structures into LLM-based models, it is important to highlight the similarities and differences between the transformer and the GCN architectures. In a comprehensive review on GNNs, Veličković (2023) argues that transformers can be considered a special class of GNNs, as the self-attention mechanism enables the transformer to capture relationships among the words in the input. For GCNs, Kipf and Welling (2016) define an adjacency matrix A, representing the connections between the nodes in a graph. Similarly, the transformer computes an attention matrix for each word in the input sequence by multiplying the key and query. Consequently, transformers can model these relationships starting from a fully connected graph structure without the need for top-down graph construction. Nevertheless, several

[3] https://github.com/scofield7419/hesyfu

studies have shown that incorporating linguistically derived dependency structures can enhance their performance on various tasks, including NLI. These results suggest that although transformers are capable of modeling relationships between words in the training data through the self-attention mechanism, the integration of such relationships from a top-down linguistic perspective can be beneficial for the task of NLI (Bai et al., 2021; Zhou et al., 2020; He et al., 2020; Xu et al., 2022).

Although enhancing BERT with syntactic structures has shown improved performance on various NLP tasks, several studies have brought to question the usefulness of enhancing language models with syntactic structures. Investigations in the embeddings of BERT reveal that Bert-based models capture constituency and grammatical knowledge during pre-training and that the linguistic structures are largely preserved after fine-tuning, suggesting that BERT may already possess some form of syntactic knowledge (Luo, 2021; Tenney et al., 2019; Zhou and Srikumar, 2022). In addition, recent studies have questioned whether integrating an inductive bias towards linguistic syntactic structures into language models is advantageous for NLI. Research on Tree-LSTMs by Shi et al. (2018) reveals that trivial tree structures such as balanced binary trees perform on par with constituency tree-LSTMs in NLI tasks. Likewise, Yu et al. (2022) found that incorporating trivial trees into RoBERTa through GCNs yielded comparable performance enhancements to using dependency trees on diverse GLUE benchmark datasets (Wang et al., 2018), with both producing marginal improvement over the baseline. Finally, Glavaš and Vulić (2021) established that BERT's IPT on Universal Dependency grammar provides minor improvements on NLI and, conversely, degrades RoBERTa's performance. They further highlighted that post-IPT training model data representations differ from those post-NLI fine-tuning, suggesting that features learned through IPT may not be beneficial for NLI. Therefore, a number of studies in the field have prompted reconsideration of the benefits of integrating syntactic structures into language models. Nevertheless, He et al. (2020) have found greater performance in enhancing BERT with dependency structures via GCNs on the out-of-domain evaluation set of HANS, indicating that different methodologies may provide different results.

2.4 Related Work on the HANS dataset

The HANS dataset evaluates whether language models have learned to use fallible and shallow lexical overlap heuristics (McCoy et al., 2019), and the deficiencies of BERT-based models on the HANS dataset have been investigated in the literature. Min et al. (2020) examined BERT's shortcomings on the HANS dataset, introducing the "Missed Connection Hypothesis" suggesting that BERT learns syntax during pre-training but lacks the syntactic signal from the training data to apply it to the NLI task. They tested this hypothesis by augmenting MNLI with 405 sentence pairs in which the subject and object in the premise swap positions in the hypothesis, which led to improved BERT performance on HANS. However, BERT still underperformed on passive subcases in HANS, supporting their "Representational

Inadequacy Hypothesis", which proposes BERT lacks robust passive syntactic representations, due to a lack of pre-training for this form, and requires more training data for the passive case to make up for this deficiency. Lastly, Wu et al. (2022) use generative AI to create new samples for SNLI and MNLI, and de-bias them using statistical methods to counter annotation artifacts. Their data augmentation experiments show a notable increase in BERT's performance on HANS when trained on the augmented MNLI dataset. Thus, existing work suggests that MNLI may not contain sufficient signal to teach the original BERT model how to use syntax for the task of NLI. Nevertheless, work by He et al. (2020) has shown increased performance for the out-of-domain HANS dataset when enhancing BERT with dependency structures, indicating that linguistically informed structures can be beneficial in learning more informed representations from the training data. However, the effects of enhancing BERT with constituency structures only, and both structures on the HANS dataset have not been sufficiently studied, a gap that we aim to address in this thesis.

2.5 Related Work on Monotonicity Reasoning

To investigate the ability of language models to perform monotonicity reasoning, Yanaka et al. (2019a,b) create the HELP dataset using rule-based methods, and MED dataset using crowd-sourcing methods. Using the MED dataset as evaluation and HELP as data augmentation, they show that models trained on MNLI struggle to model the context of monotonicity reasoning and that their performance is largely dependent on the proportion of upward and downward cases in the training set. Rozanova et al. (2022) replicate this finding, and further show that the model trained on the MNLI training set augmented with HELP is better able to distinguish features related to monotonicity context, whereas the model trained only on MNLI is unable to distinguish between upward- and downward monotonicity features.

Consequently, Chen (2021) investigate the effects of incorporating dependency structures into tree-LSTM models, and show improved performance over BERT-based models on the MED dataset, indicating that incorporating linguistic structures in neural-based language models can be beneficial for the monotonicity reasoning task for smaller, lighter-weight models. Current SOTA performance on MED is achieved by Chen et al. (2021) through their work on NeuralLog, a hybrid model that combines traditional symbolic AI through the identification of constituent polarities in the premise from Universal Dependency trees[4], and the robustness of neural networks to detect syntactic variations for the task of NLI. With an overall performance of 93.4% on the MED dataset, their results demonstrate the potential of hybrid architectures to overcome the limitations of purely neural network-based models.

Nevertheless, the effects of enhancing LLM-based models with syntactic struc-

[4] https://github.com/eric11eca/Udep2Mono

tures have not been explored for the monotonicity reasoning task. Because monotonicity reasoning requires the model to make logical inferences from the syntactic structure of a sentence, we hypothesize that enhancing a neural network with syntactic structures can be beneficial in helping the model identify the context of monotonicity reasoning, improve its performance on the MED dataset, and potentially overcome the limitations of purely neural-based models.

3 Methodology

3.1 Architectural Overview

In this thesis, we adopt the syntax-based GCN implementation by Fei et al. (2021), Hesyfu[5], which incorporates both a constituency-based GCN (ConstGCN, original work by Marcheggiani and Titov (2020)) and dependency-based GCN (DepGCN) for the task of Semantic Role Labeling. The implementation of both syntactic structures grants us the opportunity to research the effects of each individual syntactic structure, as well as the combination of both structures. As the architecture has not been developed for the task of NLI, we pick parts of the implementation by He et al. (2020) related to aligning the premise and hypothesis to build a coherent pipeline for the NLI task. We rename Hesyfu to DC, constGCN to Con, and depGCN to Dep in this thesis for simplicity reasons.

3.1.1 BERT as Baseline Model

To limit the computational costs of the experiments, we choose "BERT-base-uncased" as our baseline model and as the backbone of the enhanced models (Devlin et al., 2018). BERT encompasses the fundamental components inherent in LLMs, possesses bidirectional contextual understanding capabilities required for the NLI task, and is designed to efficiently represent input data as pairs of sentences. Sentences are pre-processed and tokenized by concatenating them as "[CLS] p [SEP] h, [SEP]", where p and h equal the premise and hypothesis respectively and [CLS] and [SEP] are BERT specific tokens indicating task and end of sentence respectively. For the base model, we perform classification using the [CLS] token, which we use to represent the semantic representation of the sentence pair at the end of the feedforward phase. For the GCN components, we follow the work by He et al. (2020) by separating the premise and hypothesis pair for the follow-up syntax enhancements, discarding the [CLS] and [SEP] tokens in the process.

3.1.2 Constituency GCN Layer

The implementation of the constituency GCN (Con) layer succeeds the work by Marcheggiani and Titov (2020), and Fei et al. (2021) adopt it for their research. The

[5] Heterogeneous Syntax Fuser

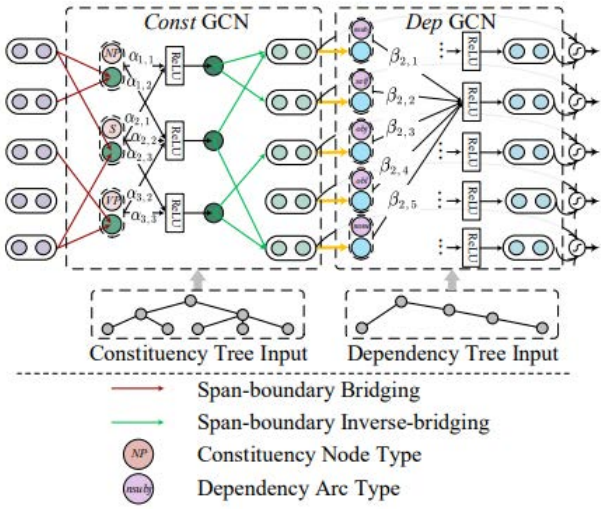

Figure 2: Heterogeneous Syntax Fuser (DC-GCN) architecture from Fei et al. (2021). sentences are passed through the GCN layers and fused through highway units to preserve information across layers (Srivastava et al., 2015).

left side of Figure 2 provides an overview of the implementation of the constGCN model. Following the work by Fei et al. (2021), given the constituent tree $G^{(c)} = (U^{(c)}, E^{(c)})$ where $U^{(c)}$ is the node set and $E^{(c)}$ the edge set, firstly, in the **Span-boundary Bridging** operation (red arrows), the non-terminal labels from the tree are concatenated to the embeddings derived from BERT. Then, the nodes $h_u^{(c)}$ are updated using the edges found in the constituency tree:

$$h_u^{(c)} = \text{ReLU}\left\{\sum_{v=1}^{M} \alpha_{uv}\left(W^{(c_1)} \cdot r_v^b + W^{(c_2)} \cdot v_v^{(c)} + b^{(c)}\right)\right\} \tag{3}$$

Where u and v represent nodes in $U^{(c)}$. r_v^b represents the initial representation of node v, which is calculated by adding the start and end token of the phrasal span: $r_v^b = r_{\text{start}} + r_{\text{end}}$. $v_v^{(c)}$ represents the embedding for the label of node v, and α_{uv} encodes both syntactic edge and label information, and represents the weights of the constituent connecting distribution:

$$\alpha_{uv} = \frac{e_{uv}^{(c)} \cdot \exp\{(z_u^{(c)})^T \cdot z_v^{(c)}\}}{\sum_{v'=1}^{M} e_{uv'}^{(c)} \cdot \exp\{(z_u^{(c)})^T \cdot z_{v'}^{(c)}\}} \tag{4}$$

Where $e_{uv}^{(c)}$ equals 1 if there is an edge between node u and v, and 0 otherwise. Note that the edges are bi-directional. $z_u^{(c)}$ is calculated as the sum of v_u and $v_u^{(c)}$.

Lastly, the **Span-boundary Inverse-bridging** operation (green arrows) is performed to restore the token node for each word w_i in the sequence $h_i^{\text{const}} = h_{u_0}^{(c)} + h_{v_0}^{(c)}$. This last operation is performed to get back the original sequence length of the sentence without the constituent nodes.

3.1.3 Dependency GCN Layer

Likewise, in the right side of Figure 2, for the dependency GCN (Dep), given the tree $G^{(d)} = (U^{(d)}, E^{(d)})$ where $U^{(d)}$ represent the node set and $E^{(d)}$ the edge set, the hidden representation $h_i^{(d)}$ is calculated as:

$$h_i^{(d)} = \text{ReLU}\left\{\sum_{j=1}^{n} \beta_{ij}\left(W^{(d_1)} \cdot r_j^{'} + W^{(d_2)} \cdot v_{ij}^{(d)} + b^{(d)}\right)\right\} \tag{5}$$

Where $r_j^{'}$ represents the initial embedding representation of token j, which is calculated as $r_j^{'} = r_j + h^{\text{const}}$ if the depGCN component follows the constGCN component, as in DC. If the depGCN component follows the BERT backbone, h^{const} will be replaced by h^{BERT}. v_{ij} represents the embedding for the dependency label between i and j, and β_{ij} encodes both syntactic edge and label information, and represents the weights of the neighbor-connecting strength distribution:

$$\beta_{ij} = \frac{e_{ij}^{(d)} \cdot \exp\left(z_i^{(d)}\right)^T \cdot z_j^{(d)}}{\sum_{j'=1}^{n} e_{ij'}^{(d)} \cdot \exp\left(z_i^{(d)}\right)^T \cdot z_{j'}^{(d)}} \tag{6}$$

$$e_{ij} = \begin{cases} 1, & \text{if } (i,j) \in \varepsilon \\ 1, & \text{if } (j,i) \in \varepsilon \\ 1, & \text{if i==j (self-loop)} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

Where ε represents the dependency edges found in the parsed tree and $z_j^{(d)} = r_i^{'} + v_{ij}^{d}$. Thus, the edges in the dependency tree are modeled as bi-directional and include self-loops.

3.1.4 Heterogeneous Syntax Fuser (DC)

For the integration of both constituency and dependency structures, both premise and hypothesis are first individually passed through the Con, then the output is passed to the Dep, and the outputs are finally fused using highway units to retain information across the GCN layers (Srivastava et al., 2015). The heterogeneous syntax fuser (DC) architecture from Fei et al. (2021) can be found in Figure 2.

The highway unit first applies a sigmoid function to the element-wise addition of the outputs of Con (h_i^{Con}) and Dep (h_i^{Dep}) to get the gated outputs g_i (8), and

then performs element-wise multiplication and addition to the gated outputs to get the fused representation s_i (9):

$$\mathbf{g_i} = \sigma(h_i^{Con}) + h_i^{Dep}) \tag{8}$$

$$\mathbf{s_i} = \mathbf{g} \odot h_i^{Con} + (\mathbf{0} - \mathbf{g}) \odot h_i^{Dep} \tag{9}$$

It should be noted that our implementation of the highway unit differs from the original approach by Srivastava et al. (2015) in that we substitute the value 1 with 0 in the equation. Srivastava et al. (2015) mention that they chose the value of 1 for simplicity reasons. During the initial architecture exploration, we found that the work by Fei et al. (2021) made use of 0 instead of 1 in the code base. We therefore tried both values to determine which would provide the best performance and ultimately opted for 0.

3.1.5 Co-Attention Layer

The outputs of the GCN components H_A and H_B, where A and B represent the premise and hypothesis, are fed to the co-attention layer (He et al., 2020). For consistency across works, we maintain the notations for the equations by He et al. (2020). We first calculate an affinity matrix C as $C = \tanh(H_A^T W_c H_B)$, which is used in the co-attention layer:

$$\mathbf{G_A} = \tanh(\mathbf{W_A H_A} + \mathbf{C}^T(\mathbf{W_B H_B})), \tag{10}$$

$$\mathbf{a_A} = \text{softmax}(\mathbf{w}_A^T \mathbf{G_A}), \tag{11}$$

$$\mathbf{G_B} = \tanh(\mathbf{W_B H_B} + \mathbf{C}(\mathbf{W_A H_A})), \tag{12}$$

$$\mathbf{a_B} = \text{softmax}(\mathbf{w}_B^T \mathbf{G_B}). \tag{13}$$

Where w and W are weight parameters, a_A and a_B attention matrices representing attention probabilities for the premise and hypothesis respectively. The output of the co-attention layer is calculated as:

$$\mathbf{h_A} = \sum_{i \in A} a_A^i \mathbf{H}_A^i, \tag{14}$$

$$\mathbf{h_B} = \sum_{j \in B} a_B^j \mathbf{H}_B^j. \tag{15}$$

The final representation is calculated and fed to a linear layer as:

$$\mathbf{y} = W[h_A, h_B, abs(h_A - h_B), h_A \odot h_B] + b \tag{16}$$

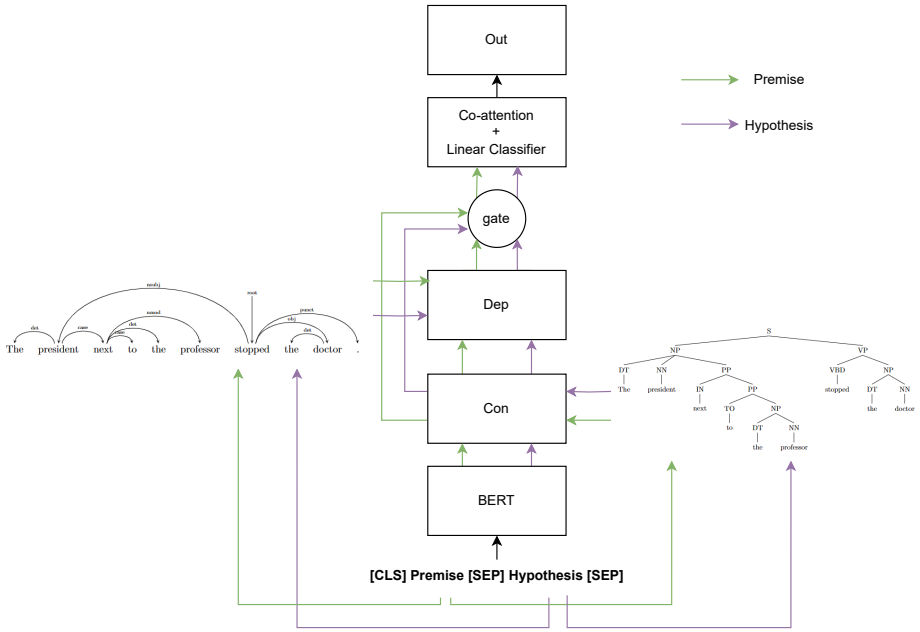

Figure 3: High-level architectural overview. The backbone of the architecture is "BERT-base-uncased". We incorporate the work by Fei et al. (2021) of label-aware syntax GCNs (Con and Dep) and methodologies from He et al. (2020) of aligning premise and hypothesis pairs for the NLI task. Left: dependency tree. Right: constituency tree.

where W equals the weights of the linear classification layer, and b the bias. We use the cross-entropy loss function to obtain the loss for the classification task. We note that the pairwise difference $abs(h_A - h_B)$ may erase some directionality effects when calculating the final representation, which can be important for NLI. However, He et al. (2020) included this in their repository, and we found through experimentation that using the absolute value provides better performance. An overview of the architecture can be found in Figure 3. For the Con-only and Dep-only variants, we omit the gating and highway functions and calculate the final representation directly using the GCN output.

3.1.6 Architectural Ordering of Components

The work by Fei et al. (2021) shows that passing the embeddings through the Con first, then dependency GCN results in better performance. Moreover, initial work on dependency GCNs by Marcheggiani and Titov (2017) shows that multiple layers of dependency GCN are needed to encode dependencies that are more than a single arc away in the dependency graph, but that by pairing the dependency GCN with an LSTM encoder allows a single GCN layer to perform better due to the LSTM

encoder being able to model global structures first. For this thesis, we maintain the same order of components as Fei et al. (2021).

3.2 From Ternary Classification to Binary

Because the HANS and MED datasets only contain two labels as opposed to three, given the output vector $\mathbf{y} = [y_e, y_c, y_n]$, where y_e, y_c, and y_n represent the output logits of *entailment*, *contradiction*, and *neutral* labels, respectively, we first apply the softmax function to obtain the probability distribution:

$$\mathbf{p} = \text{softmax}(\mathbf{y}) \tag{17}$$

Then, we compute the probability of *non-entailment* by adding the probabilities of *contradiction* and *neutral* labels:

$$p_e, p_n = \begin{cases} p_e \\ p_n & = p_n + p_c \end{cases} \tag{18}$$

This methodology differs from McCoy et al. (2019) and He et al. (2020) in that they simply translate the *contradiction* and *neutral* labels to non-entailment. In their work, McCoy et al. (2019) mention that the two methodologies provide similar results because the prediction of the model would almost always output more than 50% for the prediction, but we have found in our experiments on SNLI that adding the probabilities of contradiction and neutral provides much better results, indicating that there are cases for which models are uncertain about the probability distributions of the labels.

3.3 Syntactic Parser

To enhance the models with syntax, we parse the datasets using the Stanza parser (Qi et al., 2020), which includes both dependency and constituency parsing neural pipelines, as well as GPU acceleration. Similar to the work by He et al. (2020), we make use of Stanza's Universal Dependency parser to extract dependency features, and the default constituency model to extract constituency trees. Although SNLI, MNLI, and HANS provide constituency parses for each sentence, other datasets do not provide constituency parses. We therefore parse them once again with Stanza to ensure that our models are provided with trees from the same model. Lastly, premises and hypotheses in large-scale datasets like MNLI may contain multiple sentences. To deal with premises and hypotheses containing more than a single sentence, we set the tokenize_no_split parameter in Stanza to True. The resulting trees encompass all sentences in the premise or the hypothesis.

3.3.1 Tokenization

In this study, we observed a mismatch between the tokenization methods used by Stanza and BERT, which may lead to inconsistencies when attempting to integrate the syntactic features into the BERT embeddings. For example, consider the sentence "The man is a non-smoker." When tokenized by Stanza, the sentence is split into the following tokens: **'The', 'man', 'is', 'a', 'non-smoker', '.'.**. On the other hand, BERT employs WordPiece tokenization, which sub-tokenizes the same sentence as follows: **'the', 'man', 'is', 'a', 'non', '-', 'smoke', '##r', '.'**, sub tokenizing 'smoker' and 'non-smoker'. This discrepancy requires additional processing to align the syntactic features extracted using the Stanza parser with those derived from BERT's tokenization method. Thus, we adopt the methodology of He et al. (2020) and merge sub-tokens by mean-pooling across the sub-tokens to get the appropriate sequence lengths and align the mismatch in tokens between BERT embeddings and syntax features.

Additionally, when training our syntax models on large datasets such as MNLI, which contain long premises, we encountered issues fitting the training data into GPU memory. To deal with this issue, we individually truncate premises and hypotheses longer than 100 tokens[6].

3.4 Hyperparameters and Optimizers

For the hyperparameters and optimizers, we follow the work by He et al. (2020). For the optimizers, we make use of two optimizers during training to address the mismatch between the pre-trained BERT backbone and the newly initialized GCN encoders. To preserve the benefits of pre-training, Devlin et al. (2018) suggest that a low learning rate works best when fine-tuning pre-trained models, between 1e-5 and 5e-5. Both optimizers are initialized using AdamW (Loshchilov and Hutter, 2017). For BERT, we make use of a warmup linear scheduler. At the beginning of training, the optimizer initiates with a learning rate of zero and increases it for t warmup steps. Then, the learning rate linearly decays to zero during training. We set t to 10% of the training set. For the rest of the architecture, we make use of a multi-step learning rate. After the first epoch, we reduce the learning rate to 10% of the original learning rate.

We set the batch size at a constant of 32 for SNLI and 24 for MNLI to be able to fit the data into GPU memory[7]. The hidden dimension of the GCN layers is set to 768 to match the hidden dimensions of BERT. He et al. (2020) fine-tune their models for 3 epochs on MNLI with no early stopping. We therefore fine-tune our models for 3 epochs on MNLI, and 2 epochs on SNLI. This allows us to approximately match the number of samples seen by our models on SNLI, as SNLI is roughly 1.4 times

[6]This methodology affects 1200 sentence pairs in MNLI, which represents 0.3% of the dataset.
[7]We were unable to fit 32 batch size into GPU memory even after our truncation strategy.

larger than MNLI[8].

To find optimal learning rates, we make use of the SNLI-validation set for hyperparameter tuning. We try the following values for BERT: *5e-5*, *3e-5*, *2e-5*, *1e-5*. We then keep the learning rate for BERT fixed and tune the hyperparameters for the syntax-enhanced models. For the learning rate of the additional parts of the architecture involving GCN components, co-attention, and classification layer, we search the following values: *1e-3*, *5e-4*, *3e-4*, *1e-4*. Due to computational limits, the large size of the datasets, and the large number of LLMs to fine-tune, exhaustive or random grid-search were not viable options. We make use of the cross-entropy loss for the ternary classification task.

3.5 Dataset Shuffling

In the literature, it has been found that for the SNLI, models tend to perform best when the training data is presented in a non-shuffled format, due to the original order of SNLI presenting each premise three times, once for each label. Research by Schluter and Varab (2018) suggests that this method of presenting the dataset can increase the performance of LSTM models by 3-4% on SNLI, but leave experimentation with more complex models such as BERT to future research. We reason that, by presenting triples of premises together in a batch, BERT will be better able to separate the labels in its representation space during fine-tuning (Zhou and Srikumar, 2022), leading to increased performance. Consequently, we do not shuffle the dataset during training on SNLI-train. It is worth noting that a byproduct of this approach is the mitigation of randomness. For the experiments involving other datasets, we perform shuffling with seeding value of 42 to keep the shuffling order the same across conditions and increase reproducibility, as the datasets are not neatly sorted by premise as in SNLI, or may lack a certain label.

3.6 Normalized Subtree Kernel

For data analysis of the constituency trees, we make use of the Normalized Subtree Kernel method (Moschitti, 2006)[9]. The Normalized Subtree Kernel method computes the similarity between two parse trees by calculating the number of subtrees that they intersect and normalizing the value by the product of the number of subtrees of each tree. We first replace all words in the parsed trees with a dummy token 'x'. This procedure ensures that the similarity score will only take into consideration the tree structure and non-terminal labels, not the lexical content of the sentences. Let T_1 and T_2 be two parse trees, and let $K(T_1, T_2)$ be the tree kernel similarity between them, which is the number of common subtrees. Let $|T_1|$ and $|T_2|$ be the

[8]SNLI contains 549.361 sentence pairs after dropping missing labels, MNLI 392.702.

[9]Moschitti (2006) make the distinction between subtree kernel and subset tree kernel. Subset trees include subtrees with non-terminal symbols. In our implementation, we make use of the subtree kernel.

total number of subtrees in T_1 and T_2, respectively. The normalized tree kernel method can be defined as:

$$N(T_1, T_2) = \frac{K(T_1, T_2)}{\sqrt{|T_1| \cdot |T_2|}} \tag{19}$$

Here, $N(T_1, T_2)$ represents the Normalized Tree Kernel similarity between the two parse trees T_1 and T_2. The normalization is done by dividing the raw similarity score $K(T_1, T_2)$ by the square root of the total number of subtrees in T_1 and T_2. This normalization ensures that the similarity score lies between 0 and 1, and takes into consideration the sizes of the trees.

3.7 McNemar's Test

For the HANS and MED datasets, we perform McNemar's statistical test to determine whether the differences in performance are statistically significant for the binary classification task (McNemar, 1947). We calculate the χ^2 test statistic, which considers the null hypothesis as $H_0 : b = c$ and alternative hypothesis as $H_1 : b \neq c$, where b and c are the correct predictions made by one model and incorrect by the other. Given the following contingency table:

	Model 2 Correct	Model 2 Incorrect
Model 1 Correct	a	b
Model 1 Incorrect	c	d

Table 3: Example McNemar contingency table.

The χ^2 test statistic is calculated as:

$$\chi^2 = \frac{(b - c)^2}{b + c} \tag{20}$$

If the p-value is below 0.05, we reject the null hypothesis and therefore consider the difference in performance statistically significant. For datasets with more than two labels, like SNLI and MNLI, we do not perform McNemar's test, as one of the assumptions of this test is that the predictions are binary outcomes.

3.8 Hardware Requirements, Services, and Number of Parameters

We conduct our experiments in Google Colab[10], which provides access to GPUs for a fee, and the Dutch National Supercomputer Snellius[11], which is accessed through the HPC department at the University of Utrecht. Google Colab provides Tesla T4 GPUs as a standard, and both services provide access to A100 GPUs. For this thesis,

[10] https://colab.research.google.com/
[11] https://www.surf.nl/en/dutch-national-supercomputer-snellius

we were granted 30.000 computing resources (SBUs) for Snellius, which translates to roughly 234 hours of computing on an A100 GPU.

As each premise in SNLI and MNLI is associated with three hypotheses, one hypothesis for each label, we store the premises and the corresponding features in a dictionary to reduce RAM requirements when loading the training set of SNLI and fit the training data in a low RAM environment.

To keep track of training, hyperparameter tuning, and store models, we make use of Weights and Biases, a service for machine learning developers for end-to-end tracking of neural network models and runs[12].

Because LLMs are becoming prohibitively large and their computational and environmental costs also increase, we follow the suggestions made by Patterson et al. (2021) to include the number of parameters of our models in our work, which can be found in Table 4.

Model	**N Parameters**
Base	109.484.547
Con	115.742.655
Dep	113.072.643
DC-GCN	117.552.831

Table 4: Comparison of parameters between the proposed models.

3.9 Datasets

3.9.1 SNLI and MNLI

The datasets of SNLI and MNLI have been created to provide large-scale data for neural network-based models to be competitive in the task of NLI, with SNLI and MNLI containing 550.152[13] and 392.702 sentence pairs respectively. The dataset creations of SNLI and MNLI follow a crowdsourcing methodology, taking a premise sentence from a source, and subsequently asking crowd-workers to generate a hypothesis for each one of the three labels. In SNLI, premises originate from image captions from the Flickr30k dataset (Young et al., 2014), whereas in MNLI, premises are taken from various sources and genres of English text, as well as spoken language, making MNLI a more diverse and challenging dataset. Examples of premise-hypothesis pairs from the SNLI dataset can be found in Table 5.

[12] https://wandb.ai/site

[13] 549.361 after removing missing gold labels

Premise	Hypothesis	Label
This church choir sings to the masses as they sing joyous songs from the book at a church.	The church has cracks in the ceiling.	neutral
	The church is filled with song.	entailment
	A choir singing at a baseball game.	contradiction

Table 5: Example premise-hypothesis pairs from the SNLI dataset. Each premise is associated with at least three hypotheses. These pairs were generated through a crowd-sourcing methodology.

3.9.2 HANS

Research on annotation artifacts and hypothesis-only biases have shown that a vast majority of SNLI and MNLI[14] can be trivially solved by only considering the hypothesis and by ignoring sentence structure (Gururangan et al., 2018; Poliak et al., 2018). Consequently, McCoy et al. (2019) develop the Heuristic Analysis for NLI Systems (HANS) dataset, which targets the use of lexical overlaps, subsequence, and constituent heuristics by language models. Hypotheses in the lexical-overlap heuristic share the same lexical content as the respective premise, but with different word ordering. Hypotheses in the subsequence heuristic are fully contained within the respective premise in the same order, and in the last heuristic, hypotheses in the constituent heuristic are complete sub-trees of the respective premise. Sentences in HANS are generated using templates and are further annotated with 30 linguistic phenomena each representing a different subcase, such as passive sentences. Each heuristic is associated with 10 subcases, and each subcase is associated with 1.000 sentence pairs, resulting in a total of 30.000 sentence pairs in HANS. Lastly, samples in the HANS dataset are annotated with only two labels: entailment and non-entailment because the distinction between contradiction and neutral is unclear in many of the HANS cases. Examples of the sentences and their corresponding heuristics from the original HANS paper can be found in Table 6.

3.9.3 HELP and MED

Because the SNLI and MNLI datasets lack the use of downward monotone inferences, Yanaka et al. (2019b) develop the HELP dataset to aid in the development of neural-based language models on monotonicity reasoning. The HELP dataset contains 20.945 downward-monotone samples and 7.678 upward-monotone samples. Sentences in HELP originate from the Parallel Meaning Bank (Abzianidze et al., 2017), and new sentences are generated using rule-based replacements, making HELP a semi-automatically generated dataset. Moreover, the HELP dataset contains 6.076

[14]Hypothesis-only baseline: 69% for SNLI, 53.9% for MNLI

Heuristic	Premise	Hypothesis	Label
Lexical Overlap	The banker near the judge saw the actor.	The banker saw the actor.	E
	The doctors visited the lawyer.	The lawyer visited the doctors.	N
Subsequence	The artist and the student called the judge.	The student called the judge.	E
	The judges heard the actors resigned.	The judges heard the actors.	N
Constituent	Before the actor slept, the senator ran.	The actor slept.	E
	If the actor slept, the judge saw the artist.	The actor slept.	N

Table 6: Example sentences from HANS dataset from the original paper by McCoy et al. (2019). Models relying on heuristics in the *Heuristic* column for the Natural Language Inference task fail on Non-Entailment (N) labels.

conjunction and 438 disjunction cases targetting the use of conjunction and disjunction operators. In total, HELP contains 36K sentence pairs. Several works have used the HELP dataset to study the monotonicity reasoning capabilities of language models (Yanaka et al., 2019a; Chen, 2021; Rozanova et al., 2022). Example sentences from the HELP dataset can be found in Table 7.

Similar to the HELP dataset, Yanaka et al. (2019a) develop the MED dataset to evaluate language models on the monotonicity reasoning task. In contrast to the HELP dataset, sentences in MED are generated using crowd-sourcing methodologies, due to the semi-automatic generation of sentences in HELP sometimes resulting in unnatural sentences[15]. The MED dataset contains 1.820 upward monotone cases, 3.270 downward monotone cases, and 292 non-monotone cases, totaling 5.382 sentence pairs. Table 8 illustrates sentence pairs from the MED dataset. Similar to the HANS dataset, HELP and MED are annotated with only two labels: entailment and neutral.

3.9.4 SICK Dataset

The **Sentences Involving Compositional Knowledge** (SICK) dataset, designed as a benchmark for RTE during the SemEval-2014 challenge, was specifically created to facilitate the progress of Distributional Semantics Models (Marelli et al., 2014). As numerous linguistic phenomena unrelated to semantic compositionality - such as named entity recognition, multi-word expression, and encyclopedic knowledge - play a role in Natural Language Inference (NLI), the SICK dataset has been constructed to limit these unrelated elements, focusing solely on semantic compositionality. De-

[15] of 500 randomly sampled sentence pairs, 146 were found to be unnatural.

Section	Size	Example
Up	7784	Tom bought some Mexican sunflowers for Mary ⇒ Tom bought some flowers for Mary*
Down	21192	If there's no water, there's no whisky* ⇒ If there's no facility, there's no whisky
Non	1105	Shakespeare wrote both tragedy and comedy* ⇏ Shakespeare wrote both tragedy and drama
Conj	6076	Tom removed his glasses ⇏ Tom removed his glasses and rubbed his eyes*
Disj	438	The trees are barren ⇒ The trees are barren or bear only small fruit*

Table 7: Example sentences from the HELP dataset (Yanaka et al., 2019b). Sentences with an asterisk (*) are original sentences from the Parallel Meaning Bank (Abzianidze et al., 2017)

rived from the captions of the 8K ImageFlickr dataset (Hodosh et al., 2013) and the SemEval-2012 STS MSR-video Descriptions dataset (Agirre et al., 2012), 1500 sentences were selected and expanded upon using rules to generate multiple new sentences. These generated sentences were then paired with their original counterparts, forming premise-hypothesis pairs. Additionally, some generated sentences were matched with unrelated sentences, resulting in a total of 10,000 sentence pairs. For a more detailed insight, Table 9 exhibits example sentences from the SICK dataset.

3.9.5 ANLI, FeverNLI, WaNLI, and LingNLI

Initial work on SNLI and MNLI has opened new opportunities for neural network models to perform competitively in the field of NLI. However, the rapid pace of advancements in LLMs has quickly saturated the SOTA on the evaluation sets, and the marginal gains in performance may not be indicative of their NLU capabilities. Moreover, research on annotation artifacts has brought to question whether their performance stems from their NLU capabilities or whether they are simply modeling spurious correlations within the data distributions. Consequently, recent research has put effort in collecting more challenging datasets which aim at mitigating the biases previously found in NLI datasets, as well as providing more challenging benchmarks for NLI models. Four such datasets include Adversarial NLI (ANLI, Nie et al. 2020), Fact-Verification NLI (FeverNLI, Nie et al. 2018), Worker and AI Collaboration (WaNLI, Liu et al. 2022), and Linguist in the Loop (LingNLI, Parrish et al. 2021).

In ANLI, Mechanical Turk workers generate new hypotheses, in an adversarial setting against an NLI model. Incorrectly predicted hypotheses are verified, col-

Genre	Tags	Premise	Hypothesis	Gold
Crowd	up	There is a cat on the chair	There is a cat sleeping on the chair	NE
	up:cond	If you heard her speak English, you would take her for a native American	If you heard her speak English, you would take her for an American	E
	up:rev:conj	Dogs and cats have all the good qualities of people without at the same time possessing their weaknesses	Dogs have all the good qualities of people without at the same time possessing their weaknesses	E
	up:lex	He approached the boy reading a magazine	He approached the boy reading a book	E
	down:lex	Tom hardly ever listens to music	Tom hardly ever listens to rock 'n' roll	E
	down:conj	You don't like love stories and sad endings	You don't like love stories	NE
	down:cond	If it is fine tomorrow, we'll go on a picnic	If it is fine tomorrow in the field, we'll go on a picnic	E
	down	I never had a girlfriend before	I never had a girlfriend taller than me before	E
Paper	up:rev	Every cook who is not a tall man ran	Every cook who is not a man ran	E
	up:disj	Every man sang	Every man sang or danced	E
	up:lex:rev	None of the sopranos sang with fewer than three of the tenors	None of the sopranos sang with fewer than three of the male singers	E
	non	Exactly one man ran quickly	Exactly one man ran	NE
	down	At most three elephants are blue	At most three elephants are navy blue	E

Table 8: Example sentences from MED dataset from Yanaka et al. (2019a). Genre indicates which method was used to collect the data and tags indicate which type of monotonicity reasoning, as well as the linguistic phenomena that the sentence pairs target.

Premise	Hypothesis	Gold label
A man is jumping into an empty pool	A man is jumping into a full pool	contradiction
Children are being dressed in costumes and playing a game	Kids are being dressed in costumes and playing a game	entailment
A child is experiencing a new world	A boy under an umbrella is being held by his father who is wearing a coat dyed in blue	neutral

Table 9: Example sentences from the SICK dataset. Sentences in SICK originate from image captions, and new sentences are generated using expansion rules.

lected, and used as new training data to train a new NLI model, which is initially trained on both SNLI and MNLI. Furthermore, workers participate in three consecutive rounds, with a new NLI model trained on the adversarial data generated in the previous round, resulting in increasingly more challenging sentence pairs. The resulting dataset contains 162.865 training samples which are more challenging, and provide a new benchmark for NLU. Starting from round 2, the adversarial model is trained on the FeverNLI dataset, a dataset originally created for fact-checking and verification. The source sentences provided to the Mechanical Turk workers originate from various sources, such as the HotpotQA dataset (Yang et al., 2018), News sources from Common Crawl, fiction from StoryCloze (Mostafazadeh et al., 2016), and CBT (Hill et al., 2015).

In WaNLI, GPT-3 is used to generate sentences with similar patterns to the sentences found in MNLI. Then, human crowd-workers are employed to verify, label, and optionally revise the training data. The resulting dataset contains 107.885 sentence pairs, and initial experiments with training on the dataset show improved generalization performance on ANLI, as well as HANS.

Lastly, in LingNLI, human annotators are provided chat-room access to an expert linguist, who provides feedback, and guidance, and assesses their work during the data annotation stage. The resulting dataset contains 44.982 sentence pairs and has been found to be more challenging than MNLI, and further increases the out-of-domain generalization of models when trained on MNLI augmented with this new data.

4 Experiments

In this section, we revisit our proposed research questions and provide the details of the experiments conducted to answer them. We make the distinction between in-domain experiments where we evaluate on the evaluation set associated with the training set, out-of-domain experiments where we evaluate our models on out-of-domain distributions, and transfer-learning experiments where we first train on a generic NLI dataset, then further fine-tune on a dataset with few training samples.

4.1 Models

To explore the benefits of syntax for NLI, we conduct our experiments with four different models. We choose "BERT-base-uncased" as our baseline model, as BERT-base encompasses the primary mechanisms of pre-trained LLMs that are pertinent to our research. It is worth noting that due to computational constraints, it was not feasible to scale our models to larger variants such as BERT-large or RoBERTa within this thesis. We further include a model enhanced with dependency structures (Dep) and one enhanced with constituency structures (Con). We evaluate the performance of a model enhanced with both types of syntax (DC) to research the effects of combining both types of linguistic features. All enhanced models are derivatives of work by Fei et al. (2021) on enhancing language models with syntax GCNs.

4.2 In-Domain Experiments

[**RQ1.**] What is the effect of enhancing BERT with constituency or dependency structures with GCNs on their performance for the task of NLI?

To answer Research Question 1, we train the models on the SNLI training set and evaluate on SNLI-test, as well as SNLI-test-hard provided by Gururangan et al. (2018). SNLI-test-hard includes samples from SNLI-test that could not be solved with a hypothesis-only classifier. We further train our models on the MNLI training set and evaluate on the respective test sets, both matched and mismatched variations. MNLI-matched contains samples in the same genre as the training data: fiction, government, slate, telephone, and travel, whereas MNLI-mismatched contains similar but out-of-domain samples from different genres: face-to-face, letters, nine-eleven, Oxford-University-press, verbatim.

We follow the work by Laurer et al. (2022) and train our models on the concatenation of MNLI, ANLI, FeverNLI, WaNLI, and LingNLI, resulting in a total of 897.607 sentence pairs. Because SNLI and MNLI have been found to contain a large number of hypothesis-only and lexical overlap biases, we reason that increasing the quantity of training data may provide further insight into the role of enhancing LLMs with additional syntactic structures on generic, large-scale NLI datasets. Although MNLI has been found to contain annotation artifacts, we include the dataset in this set of experiments to provide our models with more training data and to be consistent with Laurer et al. (2022). Furthermore, hypotheses in LingNLI have been generated using the 'slate' genre in MNLI, and sentence pairs in WaNLI have been generated to follow similar patterns to samples in MNLI. Because of these similarities, including MNLI may be beneficial for these datasets. Lastly, we evaluate the models on the associated evaluation sets[16], as well as the out-of-domain evaluation sets of HANS and MED.

[16] Except the evaluation set of FeverNLI, as it misses gold labels.

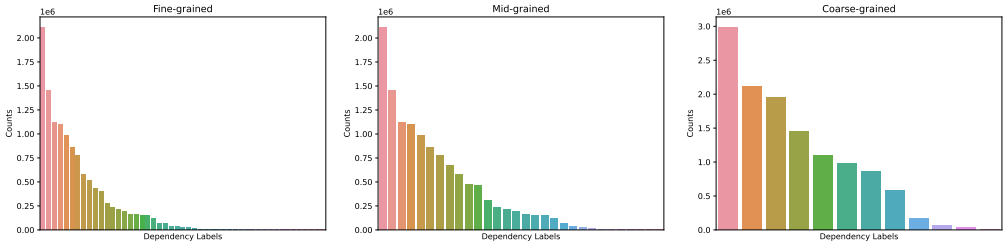

Figure 4: Dependency label frequencies in SNLI-train before and after clustering similar labels. Left: original fine-grained dependency labels. Mid: mid-grained condition. Right: coarse-grained condition.

[**RQ1.a**] Is there a beneficial effect when integrating both constituency and dependency structures for the task of NLI?

To answer this research question, we will look at our results on both in-domain evaluation sets, as well as out-of-domain evaluation sets on the HANS and MED datasets. Works in the literature incorporating both syntactic structures have reported increased performance for the task of NLI (Bai et al., 2021; Zhou et al., 2020). However, their evaluations have been restricted to in-domain evaluation sets. It therefore remains unclear whether they are still using shallow heuristics, or whether the combination of both structures aids in the monotonicity reasoning task.

[**RQ1.b**] What is the effect of fine vs coarse granularity level of the dependency labels on performance for the task of NLI?

In the literature, works on enhancing BERT with dependency structures have omitted the integration of dependency labels due to the risk of overfitting and over-parameterization. For the task of Semantic Role Labeling, Marcheggiani and Titov (2017) have argued that including dependency labels can cause models to overfit on them and consequently lower their generalization performance. Follow-up work for the task of NLI by He et al. (2020) has also chosen to omit modeling dependency labels due to the risk of overfitting the training data. However, He et al. (2020) have not experimented with implementing them in their architecture, and Fei et al. (2021) have shown by experimentation that dependency labels do improve performance on the Semantic Role Labeling task. To investigate the effects of fine- vs coarse-grained dependency labels, we choose an in-between approach by clustering low-frequency and similar dependency labels in a mid-grained and coarse-grained manner by hand-crafting clusters according to the similarity of syntactic functions they describe. The intuition behind this approach is that some dependency labels might be useful for the inference task, whereas others that occur infrequently may act as noise in the training data. Labels such as 'csubj:pass' only occur 5 times in the entire SNLI training set and learned label embeddings may not provide a good approximation for the true distribution of these labels. We thus include experiments on the Dep

model comparing less fine-grained dependency labels (Dep-midgrained), with coarse-grained dependency labels (Dep-coarse) and report results on the evaluation sets. An illustration of how the dependency label frequencies change from fine-grained to coarse-grained can be found in Figure 4, whereas the exact dependency label mappings for each condition and for each label can be found in the Appendix in Table 24.

4.3 Out-of-Domain Experiments

[**RQ2.**] Can enhancing BERT with dependency or constituency structures through GCNs help BERT generalize towards the HANS dataset?

To answer Research Question 2, we evaluate the performances of each of our models trained on SNLI and MNLI and further evaluate their performances on the HANS dataset, which targets the use of shallow heuristics based on lexical and syntactic overlaps, and further test the syntactic understanding of the models. Dependency structures have been shown in the literature to improve the generalization performance of BERT on the HANS dataset (He et al., 2020). However, the benefits of enhancing BERT with constituency structures on the HANS dataset remain unexplored. We hypothesize that similar to dependency structures, constituency structures may also help generalize towards the HANS evaluation set by informing the model about the syntactic structure of the sentences.

[**RQ3.**] Does enhancing BERT with constituency or dependency structures with GCNs help with monotonicity reasoning in the MED dataset?

To evaluate whether each type of syntax helps models identify monotonicity operators and their arguments, and thus helps the model identify the monotonicity context, we evaluate the performance of our models trained on SNLI and MNLI on the MED dataset. Because monotonicity reasoning deals with constituent replacements involving one or more words, we hypothesize that constituency structures can help the model learn useful representations for the monotonicity cases in the MED dataset, more so than dependency structures.

[**RQ3.a**] Can syntax help BERT increase performance over the baseline when trained on the monotonicity problems from HELP?

Because SNLI and MNLI contain more upward monotone cases, and further lack monotonicity operators targetted by monotonicity reasoning datasets such as MED, we include experiments where we augment the training set of SNLI with the HELP dataset in a similar fashion to Yanaka et al. (2019a), and include an additional set of experiments where we perform transfer-learning from MNLI to HELP. As data augmentation and transfer learning are two different methodologies, the outcomes of both experiments will provide further insight into the effects of enhancing BERT with syntactic structures. In doing so, we expect the syntax-enhanced models to be able to leverage the syntactic information in the HELP dataset to increase their performance on the HANS dataset. Moreover, we expect the syntax-enhanced models to be able to leverage syntax to identify the scope and

argument structure of the monotonicity operators and consequently identify the monotonicity context in MED.

[**RQ3.b**] How effective are syntax-enhanced models at identifying the scope and argument structures of conjunction and disjunction operators in the MED dataset?

The HELP dataset contains 6076 conjunction and 438 disjunction cases, which we hypothesize will help the syntax-enhanced models in learning the scope and argument structures of these operators and generalize towards the conjunction and disjunction subcases in MED. We hypothesize that the constituency-enhanced model will show improved results on the conjunction and disjunction cases, due to the operators sharing parent nodes with their arguments in the constituency tree. For the dependency-enhanced model, we may not see such improvements due to the dependency structures only forming dependency edges between two words.

4.4 Transfer Learning Experiments

[**RQ4.a**] What are the effects of enhancing BERT with syntax when transfer-learning from SNLI to the SICK dataset?

The SICK dataset contains rule-generated sentences where the syntactic structure of the sentences is important for the inference task. Furthermore, Kalouli et al. (2017) perform a manual inspection of the dataset and conclude that the inference task on SICK requires more than simple lexical semantics. In related research, Min et al. (2020) demonstrate that augmenting MNLI with only 405 sentence pairs where the subject and object are swapped in the hypothesis, leads to a 24% improvement in BERT's performance on non-entailed HANS subcases, indicating that just a small signal from the training data can teach BERT how to use syntax for the inference task. Consequently, we hypothesize that BERT trained on SICK may be able to capitalize on the syntactic signals in the SICK dataset, similar to the augmented subject/object inversion cases from Min et al. (2020). In addition, our syntax-enhanced models may be more proficient in utilizing this syntactic signal, thereby outperforming the baseline model. Because the neutral and contradiction labels in SICK and SNLI have different meanings, we do not augment by mixing SNLI with SICK[17]. Instead, we perform transfer learning by taking the best-performing models trained on SNLI, and further fine-tune them on SICK-train, which contains 4.439 training samples, and report the results on HANS and MED.

[**RQ4.b**] Can syntax help BERT adapt to a new distribution in a transfer learning and few-shot-learning setting?

Research by Laurer et al. (2022) shows that NLI datasets can be leveraged to alleviate data requirements in narrow domains. Therefore, an interesting question to explore is whether syntax can help BERT adapt to new domains when data is scarce. To answer this question, we take our models trained on MNLI and further

[17]Experiments by Bowman et al. (2015) also show that models trained on SNLI perform poorly on the neutral label in SICK, labeling them as contradiction.

fine-tune them on a small number of samples from the HANS dataset. As each of the three heuristics in HANS is annotated with 10 different subcases, for a total of 30 subcases, we sample n training samples from each subcase and use the rest of the HANS dataset as held-out test data for evaluation. This approach ensures a balanced distribution of subcases and prevents instances where certain subcases may not be allocated any samples. In this setting, we vary n from 5 and 10, resulting in a number of training samples between 150 and 300. We further take the average across 5 runs, and randomize the sampling of the n training samples.

5 Results and Discussion

In this section, we present the outcomes of our conducted experiments in accordance with the sequence of the proposed research questions (RQ1 through RQ4). We first present the findings from our in-domain experiments (RQ1.a-b). Subsequently, we shall present the results for our out-of-domain research questions on the HANS dataset (RQ2), followed by the outcomes on the MED dataset (RQ3), and answer the sub-questions regarding our experiments with the HELP dataset (RQ3.a) and an analysis of the performance on the conjunction and disjunction cases in MED (RQ3.b). Furthermore, we shall present the results derived from our transfer-learning experiments on the SICK dataset (RQ4.a) and few-shot-learning on HANS (RQ4.b). We further analyze and discuss our results by comparing them to the results reported in the literature when possible for similar conditions.

5.1 In-Domain Experiments (RQ1)

SNLI and MNLI

From the results in Table 10, we report that the performance of Con and DC are very similar to the baseline on SNLI-test and SNLI-test-hard, whereas Dep has decreased performance on the SNLI test and SNLI-test-hard. When trained on MNLI, we find a decrease in performance for all syntax models, with worse performance on the matched variation of MNLI.

For MNLI, when we analyze the performance of each model by genre in Table 11 and 12, we notice that for the matched evaluation set, there is more variance in the performance between each genre, with models performing best on government, and worst on slate, whereas for the mismatched variation, our models perform similarly between each genre.

The sentences in the 'telephone' genre contain many filler tokens, and lack the use of punctuation, for example in the premise: *"there's a uh a couple called um oh i'm going to forgot his name now uh Dirkson"*. We hypothesize that these properties may act as noise within the grammatical structure of the sentence, making it more difficult to learn useful syntactic representations from these examples.

	SNLI				MNLI			
Model	test	Δ	hard	Δ	m	Δ	mm	Δ
BERT	90.53		80.62		**84.26**		**84.26**	
Con	90.57	+0.04	**81.02**	+0.40	83.95	−0.31	84.01	−0.25
Dep	90.20	−0.33	80.25	−0.37	83.83	−0.43	84.24	−0.02
DC	**90.59**	+0.06	80.74	+0.12	83.36	−0.90	83.76	−0.50

Table 10: Performances for each model fine-tuned on SNLI or MNLI on the respective evaluation sets. Δ values represent difference in performance to baseline BERT. Syntax models performance is comparable on SNLI.

	BERT	Con	Dep	DC
fiction	0.84	0.84	0.84	0.83
government	0.88	0.87	0.87	0.87
slate	0.80	0.80	0.80	0.79
telephone	0.84	0.83	0.83	0.84
travel	0.86	0.86	0.85	0.85

Table 11: Performance by genre for each model on MNLI-matched evaluation set.

	BERT	Con	Dep	DC
facetoface	0.84	0.84	0.85	0.84
letters	0.85	0.85	0.85	0.84
nineeleven	0.84	0.85	0.84	0.84
oup	0.85	0.83	0.85	0.84
verbatim	0.83	0.83	0.83	0.82

Table 12: Performance by genre of each model on MNLI-mismatched evaluation set.

Furthermore, sentences in the slate genre originate from a magazine written between 1996 and 2000. For example, the premise: *"As long you have your own household in order, fretting about your neighbor's spending habits is a lot like fretting about the color of his living-room rug."* with the associated hypotheses:

Entailment: As long as your house is in order, worrying about your neighbors spending is useless.
Neutral: You shouldn't be so nosy with your neighbors.
Contradiction: You should worry about the color of your neighbor's rug.

We hypothesize that for the slate genre, some of the sentences may require common sense and world knowledge, and providing BERT with additional syntactic structures would not aid in learning useful representations from these cases. However, we note that the differences in performance between syntax-enhanced models and BERT-base for each genre are minimal, and further investigation is required to validate our claims.

Our results therefore suggest that enhancing BERT with syntactic structures with GCNs may not be useful for NLI, similar to findings by Glavaš and Vulić (2021). However, the similar performances of our syntax models to the baseline on the SNLI-test-hard evaluation set indicate that similar to BERT, the syntax models may be influenced by hypothesis-only biases in the training data. Research

on hypothesis-only biases by Gururangan et al. (2018) and Poliak et al. (2018) have shown that simple LSTM and bi-gram models can achieve an accuracy of up to 69% on SNLI-test and 53% on the MNLI evaluation set when only provided access to the hypothesis. Furthermore, research by Min et al. (2020) shows that by augmenting the training set of MNLI with only a small number of samples where the subject and object swap positions in the hypothesis, BERT is able to substantially increase its performance on the HANS dataset across a variety of subcases[18]. This result indicates that the original training set of MNLI may not contain a sufficiently strong syntactic signal for BERT to learn how to use syntax for the inference task. We therefore hypothesize that enhancing BERT with additional syntactic structures may not be beneficial for SNLI and MNLI, in part due to the hypothesis-only biases. In the next subsections, we explore how annotation artifacts, specifically hypothesis-only biases, may be conflicting with syntactic signals in the training data.

Investigating the Role of Hypothesis-Only Biases

Following our results and analysis of the literature, we posit that enhancing BERT with syntactic structures may not be beneficial for the original SNLI and MNLI datasets in part due to hypothesis-only biases. The hypothesis-only biases allow sentence pair classification by ignoring the premise, and without the need for syntactic understanding (Gururangan et al., 2018; Poliak et al., 2018). Could the biases be overshadowing the relevance of syntactic information for the classification task?

To answer this question, we devise an experiment in which we attempt to minimize these hypothesis-only biases in the SNLI and MNLI training data. Our approach for the data curation stage is as follows: firstly, we divide the SNLI training data into two halves, ensuring that premises unique to one half are not duplicated in the other. Each half of SNLI-train is used both as training data and as held-out evaluation data. Subsequently, we employ a bidirectional LSTM with only access to the hypothesis[19] [20]. We train the hypothesis-only model separately on each half of SNLI; first on the initial half of the dataset, and store incorrect predictions on the held-out second half at the end of training. We repeat the training process for the second half of SNLI-train with a separate hypothesis-only model, now using the first half as held-out evaluation data, and likewise storing the incorrect predictions at the end of training. At the end of both training and evaluation procedures on each half, we merge the incorrect predictions from both runs, resulting in the curation of the SNLI-train dataset. The incorrect predictions represent the set of hard cases that can not be trivially solved by a hypothesis-only model. To be consistent

[18]The best-performing condition augmented MNLI with 405 sentences, which represents 1% of the total size of MNLI.

[19]Details for hypothesis-only bi-LSTM model: 6B.100d Glove embeddings, trained for four epochs, batch size 32, learning rate 1e-3, and a warmup scheduler set at 10% warm-up steps.

[20]We employ a bi-LSTM here instead of BERT because works in the literature use weaker models than BERT (Gururangan et al., 2018; Poliak et al., 2018)

	SNLI				MNLI			
Model	test	Δ	hard	Δ	m	Δ	mm	Δ
BERT	63.73		82.00		74.24		75.04	
Con	65.19	+1.46	83.29	+1.29	74.17	−0.07	**75.46**	+0.42
Dep	65.12	+1.39	82.89	+0.89	73.44	−0.80	74.68	−0.36
DC	**65.98**	+2.25	**83.47**	+1.47	**74.40**	+0.16	75.34	+0.30

Table 13: Performances for each model trained on SNLI-train-hard or MNLI-train-hard for 2 epochs on their respective evaluation sets. Δ values represent the difference against baseline BERT.

with terminology by Gururangan et al. (2018), we name the curated SNLI-train as **SNLI-train-hard**. For MNLI, we use the same procedure to curate the training set of hypothesis-only biases, resulting in **MNLI-train-hard**.

The curated dataset of SNLI-train-hard contains 183.632 training samples and represents 33.43% of the original training data, whereas MNLI-train-hard contains 183.029 training samples, which represents 46.61% of the original MNLI training set. These proportions are roughly the proportion of hypothesis-only cases found in the respective datasets by Gururangan et al. (2018). Furthermore, the split for each genre in MNLI-train is as follows: telephone 41.131, slate 38.254, fiction 37.073, travel 33.914, and government 32.657, indicating that the government genre in the original MNLI-train dataset contains the most hypothesis-only biases, and the telephone genre the least.

Following the curation stage, in the training stage, we train BERT and the syntax-enhanced models on the curated SNLI-train-hard or MNLI-train-hard datasets for 2 epochs. This approach allows us to investigate how removing the vast majority of hypothesis-only biases in the datasets influences the effect of enhancing BERT with additional syntactic structures.

Our results on the SNLI-train-hard, as presented in Table 13, reveal that all three syntax-enhanced models exhibit improved performance on the SNLI-test and SNLI-test-hard datasets compared to baseline, indicating that the curation of the dataset has reduced the bias towards the hypothesis-only baseline. The combination of both structures displays the most improvements over the baseline. We reason that because in this condition both syntactic structures contribute to increased performance, the combination of both structures is beneficial for performance.

For MNLI-train-hard, we find that dependency structures do not help improve performance towards the evaluation sets of MNLI. For the Con and DC models, we find that the performance is similar to the baseline. However, we note that this is an improvement over our previous results, where we found a decrease in performance for all our syntax models.

Despite our results with the curated datasets, it is worth noting that during the creation of the -hard training sets, we did not remove lexical-overlap biases from the

training data, and each model performs poorly on the evaluation set of HANS (below 52%). This is because such biases necessitate interactions between the premise and the hypothesis, and are not so easily detectable. Thus, the increased performance for our syntax models over the baseline may stem from increased lexical overlap signals. Research by Sinha et al. (2021) has shown that language models such as BERT and RoBERTa are insensitive to word order shuffling when trained on MNLI, meaning that they output the same label when shuffling the position of the words in the input. Their methodology could likewise be employed in our case to test for word order sensitivity when training on the curated -hard datasets. However, we leave this exploration for future endeavors. Lastly, the overall performance on the full evaluation sets has decreased because the vast majority of the evaluation sets of SNLI and MNLI still contain hypothesis-only biases.

In conclusion to this subsection, by eliminating a vast majority of hypothesis-only biases in the training data, we demonstrate that the syntax enhancement of BERT yields greater benefits due to the hypothesis-only biases not outweighing the role of syntactic information. For the curated dataset of SNLI-train-hard, we find that enhancing BERT with both constituency and dependency structures increases performance, consequently leading to the incorporation of both syntactic structures to benefit performance.

Syntax for General Purpose NLI

Our initial results on SNLI and MNLI indicate that enhancing BERT with syntax via GCNs does not provide benefits for the task of NLI. However, SNLI and MNLI have been found in the literature to contain large amounts of hypothesis-only biases, which we have found to boost the performance of the BERT base model.

In this subsection, we experiment with the more recent and challenging NLI datasets of MNLI, FeverNLI, ANLI, WaNLI, and LingNLI, following the work by Laurer et al. (2022)[21]. These datasets employ more elaborated methodologies for data collection to limit the role of hypothesis-only and lexical overlap biases and are more challenging than SNLI and MNLI due to the adversarial data generation (ANLI), or through the employment of expert linguists during the data collection process (LingNLI). The results of these experiments will help us further understand whether enhancing BERT with syntactic structures is beneficial in generic NLI datasets when provided with large amounts of diverse training data.

The results in Table 14 show that BERT base outperforms the syntax models on three out of the seven evaluation sets (ANLI, LingNLI, HANS). For our syntax models, we find marginal improvements in performance on the evaluation sets of MNLI, and the performance on the MED evaluation set is very similar to baseline. Lastly, we find a large decrease in performance for our syntax models on the out-of-domain evaluation set of HANS, indicating that they are not learning useful

[21] https://huggingface.co/MoritzLaurer/DeBERTa-v3-large-mnli-fever-anli-ling-wanli

Model	M-m	M-mm	ANLI	WaNLI	LingNLI	HANS	MED
BERT_{MFAWL}	84.22	84.18	**48.78**	70.62	**77.70**	**72.97**	46.12
Con_{MFAWL}	84.20	**84.43**	47.81	70.40	77.13	69.08	46.10
Dep_{MFAWL}	84.33	83.98	47.25	**71.46**	77.21	70.87	46.08
DC_{MFAWL}	**84.50**	84.12	45.91	70.64	77.07	70.02	**46.15**

Table 14: Results for models trained on MNLI (M), FeverNLI (F), ANLI (A), WaNLI (W), and LingNLI (L). Evaluation sets of HANS and MED are out-of-domain.

syntactic representations, and are overfitting on the training data.

The poor performance of our syntax models on the ANLI evaluation set raises the question of whether the challenging aspects of ANLI may require other forms of knowledge, not directly linked to syntax. We have previously hypothesized based on our results on the evaluation sets of MNLI that sentences in the slate genre may require common sense or world knowledge to solve. Research by Williams et al. (2022) investigates which types of inferences the sentences in ANLI target. They annotate the development sets of ANLI and reveal that 40-60% of the examples in ANLI require numerical or common sense knowledge, and 24% of the examples require world knowledge. For the *Syntactic* category, which represents sentences that change argument order in the hypothesis, they report that only 10.5% of the development set contains sentence pairs that require syntactic knowledge, sentence pairs in the Syntactic category encompassing 14.5%, 8.0%, and 9.3% of round A1, A2, and A3 respectively. This reduction in cases in the Syntactic category from A1 to A3 indicates that the language models were able to learn the relevance of syntactic information when trained on the cases from the previous rounds.

An analysis of the performance of each of our models on the *Syntactic* tag in the annotated development set of ANLI reveals performances of 41.25%, 40.06%, 35.31%, and 40.06% accuracies for BERT, Con, Dep, and DC respectively. This result, in conjunction with the degradation in performance on the HANS dataset compared to the baseline, indicates that the syntax models are not learning to leverage syntax for the inference task, but rather, may be overfitting on the other aspects of the NLI task for which syntactic knowledge does not directly help with. We provide full fine-grained results in Table 30 in the Appendix. Furthermore, the average premise length in ANLI is 54.13, whereas for MNLI and SNLI these are 19.81 and 12.85 respectively. As ANLI contains multi-sentence premises, this increase in sequence length may pose other challenges for our syntax models, especially the constituency-enhanced model, as the constituents will be far away from the root. Recent work in the literature by Liu et al. (2023) reveals that auto-regressive LLMs perform worse as context length increases for the task of multi-document Question Answering, and key-value retrieval. Moreover, performance significantly degrades when relevant information is situated in the middle of the context, as opposed to the beginning or the end. These results and observations raise the important question

of what the effects are of long vs short sequence length on the performance of our syntax-enhanced encoder-based BERT models, and may be an important area for future research in the context of document-level NLI.

[**Answering RQ1**] *What is the effect of enhancing BERT with constituency or dependency structures with GCNs on their performance for the task of NLI?*

In conclusion to Research Question 1, our results indicate that enhancing BERT with syntax through GCNs does not improve performance for the task of NLI. Neither constituency nor dependency structures improve performance significantly over baseline. Our primary explanation is that enhancing BERT with syntax via GCNs is not beneficial for NLI. Additionally, general-purpose NLI requires understanding many other aspects of natural language, such as numerical or common sense knowledge, and enhancing BERT with syntax may overfit the models in these cases, as there may not be a direct correlation between the underlying syntactic structure of the sentences and the entailment label. However, as we have shown in our SNLI-train-hard and MNLI-train-hard experiments, removing sentences that do not require an understanding of sentence structure can increase some of the benefits of enhancing BERT with an inductive bias towards syntax.

[**Partially answering RQ1.a**] *Is there a beneficial effect when integrating both constituency and dependency structures for the task of NLI?*

Having presented the results of our in-domain experiments, we aim to partially answer RQ1.a. Our results show that combining both syntactic structures as is done in the DC model increases performance marginally on the evaluation set of SNLI and SNLI-test-hard, and decreases performance on MNLI-m and MNLI-mm. When provided with large amounts of diverse training data, DC marginally performs better than baseline on MNLI-m, and performs worse on MNLI-mm, and the more challenging datasets of ANLI and LingNLI compared to baseline. These results indicate that the incorporation of both syntactic structures does not aid in better performance for the task of NLI. Nevertheless, our results on the curated SNLI-train-hard dataset show that, when both individual structures are beneficial in modeling the training data, the combination of both structures can be beneficial for performance.

Dependency Label Granularity Results (RQ1.b)

In our experiments, we find that there is a drop in performance on in-domain data when enhancing BERT with dependency structures, whereas works in the literature report no drop in performance for BERT enhanced with dependency structures (He et al., 2020). In their works, Marcheggiani and Titov (2017) have omitted the incorporation of dependency labels due to the risk of overfitting and overparameterization

Model	n-labels	SNLI		HANS		
		test	hard	E	NE	All
Dep	49	90.20	80.25	93.03	**28.59**	**60.81**
Dep-mid	30	**90.51**	**80.68**	**95.57**	25.84	60.7
Dep-coarse	12	90.42	80.31	95.21	23.08	59.1

Table 15: Experimental results on SNLI and HANS test sets for Dep models with different levels of label granularities. The column n-labels indicate how many labels are associated with each condition. E and NE represent Entailed and Non-Entailed cases respectively.

for the task of Semantic Role Labeling, and He et al. (2020) likewise for the task of NLI. However, Fei et al. (2021) have shown improved performance for the task of Semantic Role Labeling when including the dependency labels. Consequently, we experiment with different granularity levels for the dependency labels. From the results in Table 15, we notice that clustering similar dependency labels can slightly improve the performance of the model on in-domain test data, but the performance gains are very minimal. Both Dep-mid and Dep-coarse show improved performance on the SNLI test sets, with the mid-grained condition performing best. However, from the results on HANS, we see that as the granularity of dependency labels goes from fine to coarse, the performance on non-entailment cases decreases, indicating more use of simple heuristics for the inference task.

[**Answering RQ1.b**] *What is the effect of fine vs coarse granularity level of the dependency labels on performance for the task of NLI?*

In conclusion to Research Question 1.b, our results indicate that clustering similar labels marginally improves performance on in-domain evaluation sets and decreases performance on the non-entailed cases in HANS. In the literature, He et al. (2020) omit the incorporation of dependency labels due to the risk of overparameterization. However, they do not experiment with incorporating dependency labels in their architecture. Our results show that incorporating dependency labels as is done in the Dep model can be beneficial for the task of NLI for out-of-domain datasets. Lastly, as we have found marginal performance gains for in-domain data, and decreased performance towards out-of-domain distributions, we maintain the original, fine-grained dependency labels for our subsequent experiments.

5.2 Out-of-Domain Experiments

Results on HANS (RQ2)

Our previous results have indicated that enhancing BERT with an inductive bias towards syntax via GCNs does not benefit the task of NLI. However, dependency

Model	HANS E	HANS NE	HANS All	HANS Δ
BERT_S	94.95	24.95	59.95	
Con_S	**98.99**	18.97	58.98	−1.0
Dep_S	93.03	**28.59**	**60.81**	+0.86
DC_S	98.87	17.27	58.07	−1.88
BERT_M	95.98	**16.31**	56.15	
Con_M	**97.71**	15.68	**56.70**	+0.55
Dep_M	95.52	15.31	55.42	−0.73
DC_M	96.65	15.50	56.07	−0.08
BERT_M (He et al. (2020))	99.0	16.87	57.9[22]	
BERT+SGCN_M (He et al. (2020))	97.5	23.5	60.5	+2.60
CAGCN_M (He et al. (2020))	97.77	**29.13**	**63.5**	+5.60

Table 16: Performances of models on HANS entailment (E) and non-entailment (NE) for models trained on SNLI (S) or MNLI (M). Underscored values represent statistical significance ($p<0.05$) against baseline. We include models provided by He et al. (2020). Fine-grained results for each heuristic can be found in Table 25 in the Appendix.

structures have been shown in the literature to help BERT generalize towards the out-of-domain HANS dataset (He et al., 2020). We have further hypothesized that constituency structures may provide similar benefits on the out-of-domain evaluation set of HANS, as the heuristics in the HANS dataset target the use of shallow heuristics based on lexical and syntactic overlaps. A model enhanced with constituency structures may be able to leverage the syntactic structures in the constituency trees for the inference task, consequently increasing performance on the non-entailed cases in the HANS dataset. In this subsection, we report our results on the HANS dataset.

The results in Table 16 show that when fine-tuned on SNLI, dependency structures help BERT generalize towards the HANS dataset, and we find increased performance for the non-entailed cases. However, neither constituency nor a combination of both structures improves performance on the non-entailed cases in HANS, and their performance is worse than the baseline on all three heuristics. When fine-tuned on MNLI, we find that all three syntax models perform worse on the non-entailed cases in HANS, indicating that they make more use of shallow heuristics for the inference task. The poor performance of the DC model in both conditions indicates that the combination of both syntactic structures may not be beneficial for out-of-domain generalization. We provide full fine-grained results for HANS for each heuristic in Table 25 in the Appendix.

[22] Glavaš and Vulić (2021) report 53.3% and McCoy et al. (2019) 54.6% accuracy for BERT base. This indicates that out-of-domain performance towards HANS is not stable.

When we compare our results to the literature, the main model proposed by He et al. (2020), *CA_GCN*, displays increased performance on HANS due to the incorporation of the additional "co-attention" dependency relations between the premise and hypothesis. This architectural choice is similar to concatenating the two sentences before feeding them into BERT, allowing BERT to model the dependencies between the sentence pairs, and offering the model an enhanced capacity for the NLI task. The increased performance may therefore not be directly related to enhancing BERT with dependency structures. Comparing our model with a more similar implementation, $BERT{+}SGCN_M$[23], which incorporates dependency structures into the premise and hypothesis separately in the GCN components, we observe more similar results to our own Dep_S model.

Contrarily, when we compare our dependency-enhanced model trained on MNLI (Dep_M) to the models provided by He et al. (2020) (BERT+SGCN_M and CAGCN_M), we find that dependency structures do not help our syntax model generalize towards the out-of-domain evaluation set of HANS. We have previously found that the incorporation of fine-grained dependency labels benefits out-of-domain performance on the HANS dataset, one of the key distinctions between our models. However, another difference between our models is that the dependency GCN components by He et al. (2020) are scaled to three layers, whereas we maintain a single layer in our implementation. It is therefore plausible that scaling our GCN component may provide further benefits. Nonetheless, our explorations with model architecture revealed that simply scaling the Dep layer to multiple layers resulted in a decrease in performance, possibly due to the incorporation of dependency labels into the BERT embeddings at each layer, causing overparameterization. Consequently, scaling the dependency component may require further modifications to the architecture. Despite these considerations, the development of alternative architectures remains beyond the scope of this thesis, and we accordingly designate this exploration as an area for future investigation.

Discussion on the Effect of Constituency Syntax on HANS

Our original hypothesis proposed that the integration of constituency structures would discourage the model from relying on shallow heuristics to solve the inference relation, similar to how dependency structures help BERT generalize towards the HANS dataset. However, our results point to the opposite, indicating that the model actually employs more, rather than fewer, simple heuristics both when trained on SNLI, and MNLI.

Previous studies have found that the entailment cases in SNLI tend to have shorter sequence lengths for hypotheses compared to the other two labels, with 8.8% of hypotheses' unigrams fully embedded within the corresponding premise. In

[23] Original work for SGCN by Lei et al. (2019) using an LSTM-based model as the backbone. He et al. (2020) adopt the SGCN component into a BERT-based model for comparison.

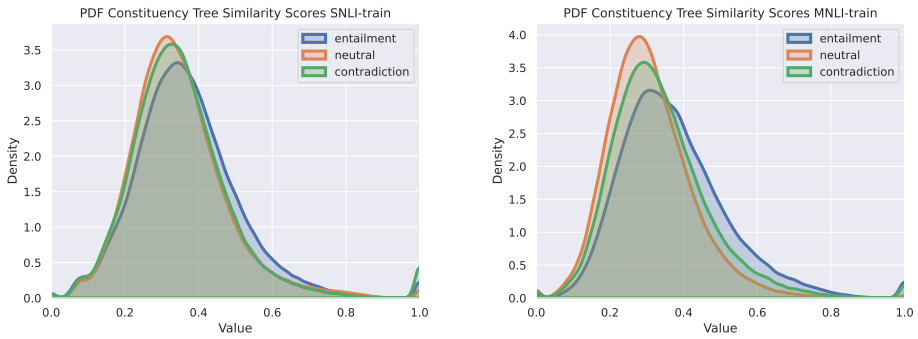

Figure 5: Probability density functions of Normalized Subtree Kernel similarity scores between premise and hypothesis. Left: SNLI. Right: MNLI. MNLI exhibits stronger syntactic overlap biases for the entailment label.

contrast, only 0.2% of instances share this characteristic for the other two labels (Gururangan et al., 2018). This disparity has led to speculation that crowd workers may use a basic strategy of removing words from the premise to swiftly generate hypotheses. Consequently, models may struggle with the heuristic challenges presented by the HANS dataset due to the lexical and subsequence overlaps for the entailment label in the training data. Given our results, and consistent with the proposed crowd-worker strategy by Gururangan et al. (2018), we propose that the sentence pairs in SNLI and MNLI also suffer from syntactic overlaps in the entailment label. Such overlaps might be exploited by a model enhanced with constituency structures, either directly, or through interaction effects with the lexical biases.

Upon examining the tree similarity density function for constituency trees in Figure 5, we find that the entailment cases in both SNLI and MNLI exhibit a slight rightward skew. This suggests that the entailed sentences in SNLI may indeed be prone to syntactic overlaps, potentially contributing to a decrease in performance when evaluated on the HANS dataset. Moreover, the average Normalized Subtree Kernel score for the SNLI training set for constituency trees for the entailment label is 0.37, whereas for the neutral and contradiction labels, the scores are both 0.35. For MNLI, the values are 0.37 for entailment, 0.31 for neutral, 0.34 for contradiction, thereby exhibiting stronger syntactic overlap biases for the entailment label.

A model enhanced with constituency structures may therefore learn these biases for the entailment label due to the embeddings in the syntax component being updated via the edges found in the constituency tree. Our results show that a model enhanced with constituency structures makes more use of shallow heuristics when trained on both SNLI and MNLI, thereby exhibiting decreased performance on the HANS dataset compared to baseline.

We note, however, that our analysis with the tree kernel similarity is limited, as we have replaced the words in the sentences with a placeholder token 'x' to isolate

Model	Up	Δ	Down	Δ	Non	Δ	All	Δ
BERT_S	**86.92**		17.46		50.68		42.75	
Con_S	86.15	−0.77	**20.83**	+3.37	51.37	+0.69	**44.57**	+1.82
Dep_S	86.04	−0.88	20.21	+2.75	51.03	+0.35	44.15	+1.4
DC_S	84.95	−1.97	19.85	+2.39	**52.05**	+1.37	43.61	+0.86
BERT_M	81.76		25.60		**54.45**		46.15	
Con_M	**82.53**	+0.77	25.11	−0.49	50.0	−4.45	45.86	−0.29
Dep_M	82.09	+0.33	**26.12**	+0.52	49.32	−5.13	**46.30**	+0.15
DC_M	81.92	+0.16	25.17	−0.43	48.97	−5.48	45.65	−0.5

Table 17: Performance on MED test set for models trained on SNLI (S) or MNLI (M). Δ values represent differences in performance against baseline BERT. Underscored values represent statistical significance ($p<0.05$) against the baseline.

the grammatical effects. Given the significant role played by a sentence's lexical contents for the semantic meaning of a sentence in the task of NLI, further research is required to substantiate this claim.

[**Answering RQ2**] *Can enhancing BERT with dependency or constituency structures through GCNs help BERT generalize towards the HANS dataset?*

In conclusion to Research Question 2, we find that enhancing BERT with dependency structures via GCNs increases out-of-domain performance on the HANS dataset when trained on SNLI, but not MNLI. Enhancing BERT with constituency structures decreases performance on the non-entailed cases in HANS in both conditions, indicating that the model may be overfitting on the lexical and syntactic overlaps in the training data. Lastly, combining both syntactic structures as is done in the DC model does not benefit out-of-domain generalization towards the HANS dataset compared to the baseline.

Results on MED (RQ3)

Our results on the MED dataset in Table 17 show slightly increased performance for our syntax-enhanced models when compared to the baseline BERT model when trained on SNLI. When trained on MNLI, we find that the syntax models perform similarly to the baseline model, and the differences in performance against the baseline model are statistically insignificant ($p>0.05$).

Interestingly, when trained on the curated SNLI-train-hard dataset, we find that Con and DC have increased performance compared to the baseline on the out-of-domain evaluation set of MED, indicating that the removal of a vast majority of hypothesis-only biases has decreased the effects of the biases in the training set, and our syntax models are better able to leverage the syntactic structures in the curated

training set for the monotonicity reasoning task. When trained on MNLI-train-hard however, we find that our syntax models have similar performance to the baseline. We provide results on MED for models trained on -hard datasets in Table 27 in the Appendix.

Our initial hypothesis posited that syntax-enhanced models, specifically Con, would demonstrate superior performance on the MED dataset because the syntactic structures would aid in distinguishing the monotonicity context from the monotone operators and syntactic structure of a sentence. Although the syntax-enhanced models do show a slight performance increase when trained on SNLI, our results are constrained by the fact that sentences in SNLI and MNLI infrequently employ monotonicity operators and they further lack examples of downward monotone cases, thereby lacking the required signals for monotonicity reasoning. Consequently, our experiments with the HELP dataset may be more indicative of the effects of enhancing BERT with additional syntactic structures.

[**Answering RQ3**] *Does enhancing BERT with constituency and dependency structures with GCNs help with monotonicity reasoning in the MED dataset?*

In conclusion to Research Question 3, our results show that providing BERT with additional syntactic structures provides slight performance gains on the monotonicity reasoning evaluation set of MED when trained on SNLI-train. When trained on MNLI-train, providing BERT with additional syntactic structures does not improve performance over baseline. However, these datasets infrequently employ monotonicity operators, and may therefore lack the required signals related to monotonicity reasoning.

Experiments with HELP Dataset (RQ3.a)

In this subsection, we shift our focus to our experiments with the HELP dataset, as we have hypothesized that the syntax-enhanced models would be able to leverage the syntactic information in HELP to increase their generalization performance towards MED[24]. We first discuss our results augmenting the SNLI training set with HELP, then the results of our transfer learning experiments from MNLI to HELP. The results of our data augmentation and transfer learning experiments with the HELP dataset can be found in Table 18.

From the results on SNLI augmented with HELP, we find that augmenting the training set with HELP has substantially increased performance on the MED evaluation set. Despite this improvement, we find that the performance of the syntax

[24]We omit the results on HANS, as HANS is unrelated to monotonicity reasoning, and the results are not informative.

[25]Models trained on $M \rightarrow H$ are not evaluated on MNLI, as the HELP dataset contains only 2 labels, and the fine-tuning process causes catastrophic forgetting of the missing label (McCloskey and Cohen, 1989)

Model	Up	Δ	Down	Δ	Non	Δ	All	Δ
BERT_{S+H}	73.74		**83.18**		53.77		**78.39**	
constGCN_{S+H}	75.60	+1.86	82.11	−1.07	53.77	0	78.37	−0.02
depGCN_{S+H}	**75.66**	+1.92	80.31	−2.87	53.42	−0.35	77.28	−1.11
Hesyfu_{S+H}	74.73	+0.99	78.99	−4.19	54.45	+0.68	76.22	−2.17
$\text{BERT}_{M\to H}$	69.40		83.64		53.08		77.16	
$\text{Con}_{M\to H}$	**72.86**	+3.46	83.64	0	60.62	+7.54	**78.74**	+1.58
$\text{Dep}_{M\to H}$	67.64	−1.76	**84.22**	+0.58	**61.30**	+8.22	77.37	+0.21
$\text{DC}_{M\to H}$	67.25	−2.15	83.09	−0.55	56.16	3.08	76.27	−0.89

Table 18: Performance on MED evaluation set for models trained on the SNLI+HELP (S+H) augmented dataset, and MNLI further fine-tuned on HELP ($M \to H$)[25]. Δ values represent performance difference against the baseline. Underscored values represent statistical significance ($p<0.05$) against the baseline. The performance of Con against BERT on MED dataset is statistically insignificant ($p>0.95$).

models is similar to baseline BERT, indicating that there may be no benefit in enhancing BERT with syntax for the monotonicity task when the model is provided access to sufficient training samples. However, calculating the agreement rates between syntax models and BERT when trained on SNLI reveal that each syntax model agrees roughly 94% with BERT base on the MED dataset. When trained on SNLI+HELP, the agreement rate on MED drops to 92%, indicating that the syntax models may be learning different representations for the monotonicity reasoning task when provided additional monotonicity samples from HELP. Our results therefore suggest that enhancing BERT with syntactic structures may not be beneficial for the monotonicity reasoning task. However, the HELP dataset only contains 36k sentence pairs, whereas SNLI contains 550k. Consequently, we hypothesize that this discrepancy in the number of samples may overshadow the syntactic signals from the HELP dataset.

Our results with transfer learning from MNLI to HELP show that the Con model performs best on MED, whereas Dep performs similarly to the baseline, and the combination of both structures decreases performance. We have initially hypothesized that enhancing BERT with constituency structures would help in learning useful representations for monotonicity reasoning, more so than dependency structures due to monotonicity reasoning dealing with constituent replacements, and additionally requires the identification of the polarities of the arguments of the monotonicity operator based on the syntactic structure of the sentence. Our results show that constituency structures indeed help our model learn useful representations from the monotonicity-driven dataset of HELP, whereas our dependency-enhanced model performs similarly to the baseline.

[**Answering RQ3.a**] *Can syntax help BERT increase performance over the baseline when trained on the monotonicity problems from HELP?*

In conclusion to Research Question 3.a, our results indicate that when augmenting the training set of SNLI with the HELP dataset, which contains monotonicity reasoning samples, enhancing BERT with additional syntactic structures does not aid in better performance on the HANS and MED datasets. When transfer learning instead of augmenting on the HELP dataset, we find that constituency structures aid in learning useful representations for the monotonicity reasoning task, whereas dependency structures and the combination of both structures do not show improvements over the baseline for monotonicity reasoning in the MED evaluation set. This result indicates that enhancing BERT with constituency structures can help the model learn more informed representations from the monotonicity-driven problems from the HELP dataset.

Analysis on Conjunction and Disjunction Cases (RQ3.b)

As the HELP dataset contains a large number of conjunction cases (6076) and a relatively small number of disjunction cases (438), comparing results on these subcases in different training conditions may provide further insight into our previous findings. We have further hypothesized that constituency structures may help BERT identify the scope and argument structure of the operators to increase the performance on the conjunction and disjunction cases, as the operators share parent nodes with their arguments in the syntactic tree.

Results in Table 19 show the performance of each model on the conjunction and disjunction cases in MED when trained on SNLI, when trained on SNLI augmented with HELP, and when trained on MNLI and fine-tuned on HELP. When trained on SNLI and SNLI augmented with HELP, we find that the performance differences for the syntax-enhanced models are relatively small. When fine-tuned on HELP, we find that our syntax models perform worse than baseline. However, the differences in performance between our syntax-enhanced models and BERT-base are statistically insignificant in every condition ($p>0.05$) due to the evaluation set of MED containing only 283 conjunction and 254 disjunction samples.

[**Answering RQ3.b**] *How effective are syntax-enhanced models at identifying the scope and argument structures of conjunction and disjunction operators in the MED dataset?*

In conclusion to Research Question 3.b, we find that enhancing BERT with syntax does not improve performance significantly on conjunction and disjunction cases, and providing additional training samples does not improve performance significantly over baseline. The primary explanation is that additional syntax does not aid BERT in identifying the argument structure of these operators. However,

Train	subcase	BERT	Con	Δ	Dep	Δ	DC	Δ
S	conj	57.84	**59.92**	+2.08	58.03	+0.19	59.16	+1.32
	disj	45.81	**47.17**	+1.36	45.72	−0.09	46.29	+0.48
S+H	conj	78.80	**78.98**	+0.18	77.94	−0.86	74.45	−4.35
	disj	48.69	**51.29**	+2.60	51.03	+2.34	47.92	−0.77
$M \rightarrow H$	conj	72.95	72.10	−0.85	**73.14**	+0.19	71.16	−1.79
	disj	**56.14**	52.47	−6.33	52.02	−5.88	50.07	−6.07

Table 19: Average performance on MED conjunction (n=283) and disjunction (n=254) cases for each model trained on SNLI (S), SNLI+HELP (S+H), and transfer learning from MNLI to HELP ($M \rightarrow H$). Upward and downward monotone cases have been aggregated in this table. Values in parentheses indicate the difference compared to baseline. The differences in performance against baseline are statistically insignificant (p>0.05).

according to research by Min et al. (2020), BERT may have already learned the syntactic structure of these operators during pre-training, and may already be able to leverage the syntactic structure of these operators for the inference task if given sufficient signal from the training data. Therefore, the performance of each model may be limited by their ability to identify whether the constituent replacement is a more generic concept or a more specific one, for which syntax may not directly help with.

5.3 Transfer-Learning Experiments

Fine-Tuning on SICK (RQ4a)

Our results from the main experiments suggest that syntax may not serve as a beneficial feature for NLI. In this subsection, we provide the results of our experiments with the semi-automatically generated SICK dataset, which has been found in the literature to require more than simple lexical semantics to solve (Kalouli et al., 2017), and limits the occurrence of linguistic phenomena unrelated to compositionality. A noteworthy distinction between SNLI and SICK is that the neutral and contradiction labels have different meanings in the datasets, and data augmentation is therefore not applicable due to this mismatch[26]. Moreover, the SICK training set is significantly smaller in size. Consequently, we perform transfer learning by first training our models on SNLI-train, then fine-tune them on the SICK training set. For this set of experiments, we evaluate on the in-domain evaluation set of SICK-test, and the out-of-domain evaluation sets of HANS and MED.

[26] Bowman et al. (2015) demonstrate that models trained on SNLI tend to label neutral cases as contradiction when evaluated on SICK.

	SICK-test		HANS			
Model	All	Δ	E	NE	All	Δ
$\text{BERT}_{S \to SICK}$	89.73		90.95	34.92	62.94	
$\text{Con}_{S \to SICK}$	**90.58**	+0.85	93.10	37.09	65.10	+2.16
$\text{Dep}_{S \to SICK}$	89.81	+0.08	82.73	**48.43**	**65.58**	+2.64
$\text{DC}_{S \to SICK}$	90.10	+0.37	**94.09**	34.33	64.21	+1.27

Table 20: Performances of models on SICK-test, and HANS entailment (E) and non-entailment (NE) for models trained on SNLI, further fine-tuned on SICK ($S \to SICK$). Δ values represent the difference in performance to baseline. Underscored values represent statistical significance ($p<0.05$) against baseline. Fine-grained results for each heuristic can be found in table 28 in the Appendix.

Our results in Table 20 show that Con outperforms baseline BERT on the SICK-test evaluation set, whereas Dep displays similar performance, and DC performs in-between the two individual syntax models. These results indicate that enhancing BERT with an inductive bias towards constituency structures can increase performance over the baseline when the underlying syntactic structure of the sentences is important for the inference task.

Our results on the HANS dataset show that further fine-tuning the models on the SICK training set improves the performance of the enhanced models compared to the baseline on the non-entailed cases. Similar to our previous results on SNLI, the Dep model performs best on the non-entailed cases in HANS, further indicating that dependency structures help the model learn useful syntactic representations for out-of-domain generalization towards the HANS dataset. We also find that Con performs better than the baseline on HANS, with improved performance on both entailed and non-entailed cases. Lastly, DC performs worse than the individual syntax models, further indicating that the combination of syntactic structures does not benefit out-of-domain generalization for the task of NLI.

In the literature, data augmentation experiments by Min et al. (2020) show that by augmenting the training set of MNLI with 405 synthetically generated subject/object inversion cases, BERT is able to improve its performance over the BERT model trained only on MNLI across many subcases, indicating that BERT is able to learn the relevance of syntactic information for the NLI task when provided sufficient signal from the training data. As the syntactic structure of the sentences in the SICK dataset is important for the inference task, in line with their work, our results show that the baseline BERT model is able to perform better on HANS when provided sufficient signal from the training data, and we further observe an increase in performance for the syntax-enhanced models, indicating that the enhanced models are better able to leverage the syntactic information in the training data to generalize towards the templates of HANS.

Our results on MED in Table 21 reveal that the enhanced models have an overall

	Up	Δ	Down	Δ	Non	Δ	All	Δ
$\text{BERT}_{S \to SICK}$	**86.04**		16.15		48.29		41.53	
$\text{Const}_{S \to SICK}$	81.92	−4.12	**39.54**	+23.39	47.95	−0.34	**54.33**	+12.8
$\text{Dep}_{S \to SICK}$	84.95	−1.09	19.42	+3.27	45.55	−2.74	43.00	+1.47
$\text{DC}_{S \to SICK}$	83.63	−2.41	27.95	+11.8	**50.34**	+2.05	47.99	+6.46

Table 21: Performance on MED dataset for models trained on SNLI and further fine-tuned on SICK ($S \to SICK$). Δ values represent difference in performance to baseline BERT. Underscored values represent statistical significance ($p<0.05$) against baseline.

increased performance over baseline when fine-tuned on SICK. The Con variant has the largest increase in performance on downward monotone cases, indicating that constituency syntax can help BERT in identifying the monotonicity context when there is a sufficiently strong syntactic signal from the training set. Dep performs similarly to baseline and we find only a small increase in overall performance on the MED evaluation set. DC performs in between the Con and Dep variants, further indicating that combining the two syntactic structures does not aid in learning good representations for monotonicity reasoning compared to enhancing BERT with only constituency structures.

It is worth noting that the performance on the MED dataset is dependent on the model being able to discriminate between upward and downward monotonicity context (Yanaka et al., 2019a; Rozanova et al., 2022), and that the SICK dataset has not been created specifically for this purpose. However, three of the sentence expansion rules in SICK deal with word replacements. Furthermore, identifying the monotonicity context is dependent on the model being able to identify the operator and the polarity of its arguments from the syntactic structure of the sentence (Yanaka et al., 2019a). Filtering the SICK dataset for downward monotone operators from Table 2 reveals that 10.48% of premises and 11.42% of hypotheses in SICK contain a downward operator, whereas only 1.75% of SNLI premises and 2.51% of hypotheses contain a downward operator, which may also explain why we only see a slight performance increase for the Con model when trained only on SNLI. The improved performance suggests that the Con model learns to leverage syntax to identify the arguments of the downward monotone operators, helping the model determine the monotonicity context. Consistent with the findings of Yanaka et al. (2019a), there is a trade-off of performance between the upward and downward monotonicity, and the addition of syntax does not alleviate this shortcoming for neural networks.

Analysis of Passive Subcases

An interesting category to explore in HANS is the passive subcase, for which BERT has been reported to perform very poorly by Min et al. (2020). One of the sentence expansion rules used in the SICK dataset turns 303 active sentences into passive

	Passive			
Model	E	Δ	NE	Δ
BERT_S	86.2		3.4	
Con_S	**99.9**	+13.79	**11.9**	+8.5
Dep_S	83.6	−2.6	7.4	+4.0
DC_S	95.9	+9.7	1.6	−1.8
$\text{BERT}_{S\rightarrow SICK}$	88.0		4.7	
$\text{Con}_{S\rightarrow SICK}$	**99.2**	+11.2	**34.5**	+29.8
$\text{Dep}_{S\rightarrow SICK}$	60.4	−27.6	11.4	+6.7
$\text{DC}_{S\rightarrow SICK}$	90.0	+2.0	4.0	−0.7
BERT_{M+MI}(Min et al., 2020)	67.0		29.0	
CA_GCN_M(He et al., 2020)	–		11.1	

Table 22: Results for passive subcases in HANS, entailed (E) and non-entailed (NE) for models trained on SNLI (S) or SNLI further fine-tuned on SICK ($S \rightarrow SICK$). Model by Min et al. (2020) has been trained on MNLI (M) augmented with 405 synthetic subject/object swap cases (MI). Δ values represent difference in performance to baseline. Underscored values represent statistical significance ($p<0.05$) against the baseline.

form, therefore making it an interesting case to analyze whether enhancing BERT with syntax helps learn this type of sentence construction when supplemented with additional training samples. In this subsection, we analyze our model's performances on the passive subcases within the HANS dataset, both when only trained on SNLI, and when further fine-tuned on SICK. The results for the passive subcategory can be found in Table 22.

Experiments by Min et al. (2020) show that augmenting with subject/object inversion cases diminishes performance on the entailed cases, and they further report that directly augmenting the training data with passive cases does not improve performance for the passive subcategory. Both results support their Representational Inadequacy Hypothesis, which proposes that BERT's pre-training phase lacks robust syntactic training for passive sentence constructions. They propose that BERT must learn the syntactic structure of passive sentences from scratch, and to overcome this limitation, a substantial quantity of passive sentences should be provided.

Our results reveal that the Con model outperforms baseline for passive cases, covering both entailed and non-entailed subcases. Additionally fine-tuning on the SICK dataset further bolsters Con's performance on non-entailed passive subcases, with little downgrade in performance on the entailed cases.

When it comes to dependency structures, we observe a slight improvement in non-entailed cases but a decrease in performance for the entailed cases. This aligns with the results reported by He et al. (2020) in relation to the passive category, as they find that their proposed dependency-enhanced model (CA_GCN) also performs

poorly on the passive subcase.

Lastly, the integration of both structures in DC slightly improves performance on entailed cases but lowers performance on non-entailed cases, indicating that the combination of both structures does not aid in learning better representations for the passive subcategory.

[**Answering RQ4.a**] *What are the effects of enhancing BERT with syntax when transfer-learning from SNLI to the SICK dataset?*

In conclusion to Research Question 4.a, our results show that both Dep and Con are better able to leverage the syntactic signals in the SICK dataset to generalize towards the out-of-domain evaluation set of HANS, with our dependency-enhanced model showing greater improvements over the non-entailed cases in HANS. Furthermore, we find that constituency structures help BERT learn more useful representations for monotonicity reasoning in the MED evaluation set, whereas for dependency structures, we find smaller improvements. Likewise, we find that despite passive sentences still being challenging for BERT, constituency structures can support the model in learning this type of sentence construction, whereas for dependency structures and the incorporation of both structures, we find no such improvements. Lastly, the DC model performs worse than Con and Dep on HANS, and worse than Con on MED and SICK-test, indicating that combining both structures is not beneficial for the task of NLI.

Few Shot Learning on HANS (RQ4.b)

Our results so far have indicated that enhancing BERT with syntactic structures via GCNs does not help on generic large-scale NLI datasets, but can provide improvements on datasets where the underlying syntactic structure of the sentences is important for the inference task. As research by Laurer et al. (2022) has shown, deep transfer learning from NLI to narrow domains can reduce the data requirements up to tenfold. Following their results, we have posed the question of whether syntactic information can help models adapt to new dataset distributions when provided with only a small number of training samples. For this set of experiments, we perform transfer learning by taking our models trained on MNLI, sample n samples from each of the 30 subcases in the HANS evaluation set, and use the rest of the evaluation set as held-out test data.

From the results in Figure 6, we observe that our syntax models have increased performance against baseline when provided with only a few training samples, indicating that models informed with syntax can better adapt towards new dataset distributions when data is scarce. Both dependency and constituency structures increase performance over baseline on the HANS dataset when trained on only a few examples from the dataset, and the combination of both syntactic features displays the most improvements over the baseline. Similar to our experiments with

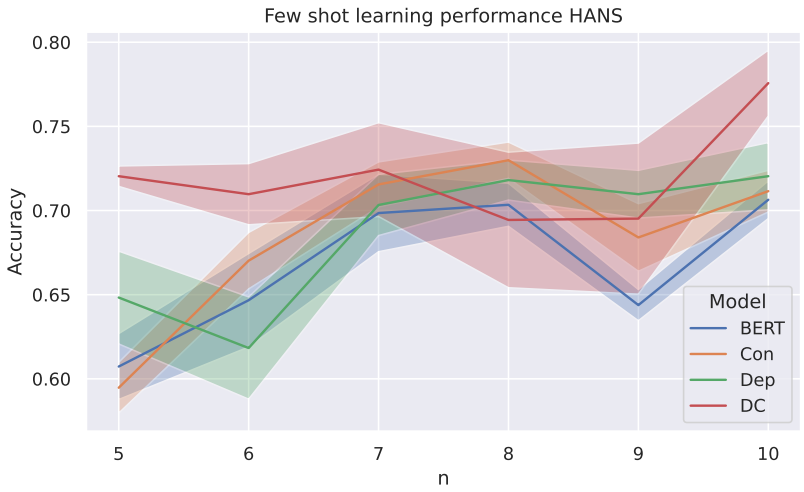

Figure 6: Performance of each model on the HANS dataset when provided with only a few training samples. N represents the number of training samples provided for each of the 30 subcases. Accuracy is calculated as the average across 5 runs.

SNLI-train-hard, the combination of both methods further increases performance. Consequently, our results suggest that when both constituency and dependency structures provide benefits individually, the combination of both syntactic structures can be beneficial.

[**Answering RQ4.b**] *Can syntax help BERT adapt to a new distribution in a transfer learning and few-shot-learning setting?*

In conclusion to Research Question 4.b, our results with fine-tuning on a small number of samples from the HANS dataset indicate that syntax helps BERT adapt to new distributions when data is scarce. As the underlying syntactic structure of the sentences is important for the inference task in the HANS dataset, this is in line with the results of our previous experiments with the SICK and HELP datasets, and we further find the additional syntax enhancements to be beneficial in settings where the size of the dataset may be limited, a common problem in narrow domains.

5.4 Discussion on Integrating Both Types of Syntactic Structures (RQ1.a)

We have previously provided a partial answer to research question 1.a based on the results of our in-domain experiments. Our in-domain experiments have shown that DC has similar performance to base BERT on the evaluation sets of SNLI, and decreased performance on MNLI. Moreover, DC performs worse than the baseline

on MNLI-mm, ANLI, and LingNLI when provided with large amounts of diverse training data. Moreover, we find that DC consistently shows decreased performance compared to Con and Dep when evaluated on the out-of-domain distributions of HANS and MED, indicating that the combination of both structures overfits the model on the training data[27]. Furthermore, our transfer-learning experiments show that DC performs worse than each individual syntax model on the non-entailed cases in HANS, and the monotonicity reasoning cases in MED, showing that combining both structures does not aid in learning useful representations for monotonicity reasoning and out-of-domain generalization.

Nevertheless, our results on the curated SNLI-train-hard dataset show that, when both individual structures provide improvements over the baseline, the integration of both syntactic structures can be beneficial for performance. Likewise, the results of our few-shot learning experiments on the HANS dataset show that when both dependency and constituency structures are useful for the inference task, combining both syntactic structures can be beneficial for in-domain performance. However, we note that the experimental settings are highly specific, and the diversity of the HANS dataset is limited due to the template-based generation process.

In conclusion to Research Questions 1.a, although works in the literature have argued that the combination of both syntactic structures via GCNs is mutually beneficial for Semantic Role Labeling (Fei et al., 2021), we find that for the task of NLI, combining dependency and constituency structures in DC does not aid in better performance for the task of NLI. Nevertheless, works in the literature have shown an increase in in-domain performance for the task of NLI when combining both syntactic structures using other methods (Bai et al., 2021; Zhou et al., 2020). Therefore, our results may be a consequence of our chosen architecture and methodology, as the DC model passes the representations of BERT first through the constGCN component, then the depGCN component. Nonetheless, the investigation of alternative architectures and methodologies for integrating syntactic structures falls beyond the scope of this thesis. Consequently, the question of whether alternative approaches for incorporating multiple forms of syntactic information yield similar outcomes remains an open avenue for future research.

6 Limitations

Having presented and discussed our results, in this subsection, we address the limitations of our research. Our experiments show that enhancing BERT with syntax via GCNs does not benefit the task of NLI on generic large-scale NLI datasets. Our chosen methodology attempts to learn useful syntactic representations directly from the training data through fine-tuning and repurposing the representations of a pre-trained model. Nonetheless, our experimental results show that eliminating a vast

[27] Accuracy on the training set is slightly higher than the baseline, and loss is slightly lower.

majority of hypothesis-only biases in the training data increases the role of syntactic information for the inference task. Consequently, we reason that this method of enhancing BERT with syntax may be susceptible to spurious correlations such as hypothesis-only biases. Additionally, our results with large-scale NLI datasets indicate that enhancing BERT with syntax via GCNs may overfit the model on other aspects of NLI which do not require syntax to solve, such as common sense and world knowledge.

Kulmizev and Nivre (2022) argue that careful consideration should be put into what kind of syntactic representations the language model is imbued with, as in order to answer whether syntax itself is useful for NLU, the type of syntax and the way in which it is imbued into the model matters for the interpretation of the results. As we have shown through the annotated development set of ANLI, when fine-tuned on general-purpose NLI datasets, our syntax models perform worse than baseline in the *Syntactic* category, as well as the HANS evaluation set, indicating that the learned representations may not be related to the understanding of sentence structure. The methodology employed by Glavaš and Vulić (2021) of Intermediate Pre-training (IPT) may address this issue by providing syntactic pre-training separately, thereby forming more linguistically accurate syntactic representations useful across many NLP domains, and by showing SOTA performance for the models as dependency parsers. Moreover, they make use of adapter-based IPT, which adds specific tunable parameters to their model, while keeping BERT layers frozen during the IPT phase. However, in the fine-tuning stage on MNLI, they fine-tune both the BERT and the adapter parameters. Similar to our approach with GCNs, this approach may allow the syntax-specific adapter parameters to adapt to the spurious correlations within the training data. Consequently, we are limited in drawing strong conclusions about whether syntax itself helps NLU, as there is insufficient evidence that our models have indeed learned syntax through the fine-tuning process on NLI datasets.

Furthermore, Kulmizev and Nivre (2022) argue that the results of fine-tuning on NLU datasets are sabotaged by the uncertainty of which specific linguistic phenomena are being evaluated by the benchmark dataset. For general-purpose datasets such as ANLI, fine-grained annotations have been limited to the development set due to the high cost of employing expert annotators. Evaluation sets such as HANS and MED, and semi-synthetically generated datasets such as SICK and HELP alleviate some aspects of this shortcoming. However, the size of MED is small, consequently rendering the differences in performance for our models statistically insignificant in the conjunction and disjunction cases. Furthermore, the sentences have been generated through templates and rule-based transformations, limiting the number of errors and diversity we would normally expect from natural language data. The lack of fine-grained annotations for large-scale datasets thus limits the extent to which we are able to analyze and interpret the results for general-purpose NLI with respect to fine-grained linguistic phenomena.

Since the inception of BERT, new methods and architectures have been pro-

posed which improve upon various aspects of the model, such as RoBERTa (Liu et al., 2019) and DeBERTa (He et al., 2021), both of which employ more robust pre-training regimes to increase generalization performance. Moreover, scaling the models has been shown to increase the performance of LLMs across many NLP tasks. For this thesis, we employed the *BERT-base-uncased* model as the backbone of our architectures. According to the Representational Inadequacy Hypothesis (Min et al., 2020), BERT has not received sufficient pre-training for the passive subcase, thereby exhibiting shortcomings for this subcase on the HANS dataset. Our results show that constituency structures can help BERT learn this type of sentence construction when fine-tuned on the SICK dataset, which targets passive sentences. However, these results may not translate to models which have been provided with more robust pre-training regimes such as RoBERTa. Due to computational limitations and the already large number of LLMs to fine-tune, we were unable to scale our models to investigate the effects of enhancing larger, or different versions of LLMs with syntactic information.

Lastly, for this thesis, we were granted 30.000 computing resources, which translates to roughly 234 computing hours on an A100 GPU. However, we were initially only granted 10.000 resources, which put a constraint on the types of experiments we were able to run. Because of this, we originally performed some experiments under subpar conditions, which we later corrected when granted sufficient computing resources. Nevertheless, with unrestricted allocation of computational resources, we would have been able to perform more exhaustive experiments, thereby increasing the robustness of our results.

Quality of Syntactic Parses

Due to the extensive size of the datasets, we employed Stanza's GPU accelerator for extracting both constituency and dependency trees. Upon analyzing the results on HANS, we observed some inaccurately parsed examples, such as the one depicted in Figure 7. Given that Stanza relies on neural models for sentence parsing, we reasoned that these inaccuracies stem from imprecise GPU floating-point arithmetic. Analyzing the dependency trees in the HANS validation set without GPU acceleration revealed 7.16% unequal edges and 7.82% unequal labels in the premises. Nevertheless, when evaluated on these CPU-parsed examples, the Dep model exhibited only a marginal 0.01% increase in accuracy on the HANS validation set, indicating that the effect of CPU vs GPU trees on performance is minimal. We note that our analysis is based on the inequality between CPU and GPU trees and that due to the lack of gold parses, we are unable to precisely investigate how many trees are wrong.

Likewise, for the constituency trees in HANS, we also observed some wrongly parsed trees. An example of an erroneous parse tree for the subject/object swap case can be found in Figure 8. Comparing each constituency tree within a subcase to each other using the Normalized Subtree Kernel score in table 23, we find that not all

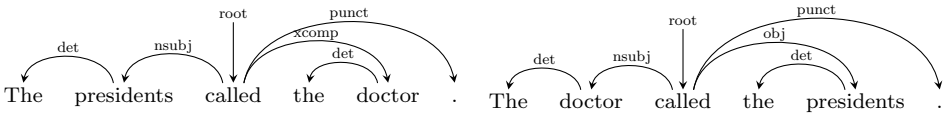

Figure 7: Example of wrong dependency tree in the HANS dataset by Stanza's universal dependency parser. Left: premise, right: hypothesis. In the left tree, the dependency relation between "doctor" and "called" is wrongfully labeled as "xcomp".

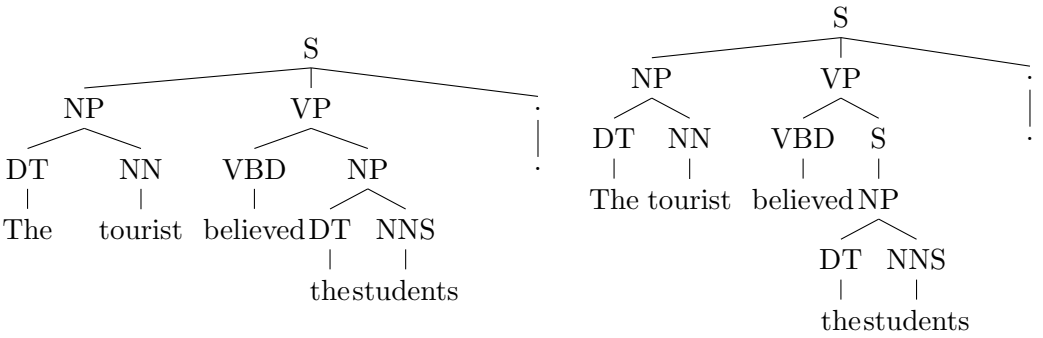

Figure 8: Left: original HANS constituency tree. Right: Example of wrong constituency treeby Stanza. An additional S node has been added to the tree, above the NP constituent. The Normalized Subtree Kernel similarity score between the two trees is 0.7.

subcases that use the same template have the same constituency parse tree, whereas we would expect their similarity score to be 1[28]. The values in table 23 indicate that roughly 2.88% of the parses may be wrong and slightly noisy. Nevertheless, when evaluating our Con model using the original HANS trees, we found no difference in performance on the HANS evaluation set. For completeness, we provide the full table of similarity scores within each subcase in the Appendix in table 29, including for the original parses provided with the HANS dataset.

According to research by Sachan et al. (2021) on enhancing BERT with dependency structures, performance for the tasks of Semantic Role Labeling and Relation Extraction in syntax-enhanced neural networks are contingent on the availability of gold, human-annotated parses, which were available for their tasks. However, due to the absence of gold parses for NLI datasets, we are limited in examining the potential impact of training on gold parses for the training sets on the performance of syntax-enhanced language models. It is possible that providing BERT with gold parses during training may increase performance for the enhanced models. However,

[28]Some subcases in HANS have multiple templates, and the similarity score within the subcase will therefore not be one.

	Original HANS		Stanza	
Subcase	Premise	Hypothesis	Premise	Hypothesis
ln_subject/object_swap	1.0	1.0	0.95	0.95
ln_passive	1.0	1.0	1.0	0.94
le_passive	1.0	1.0	1.0	0.94
se_understood_object	0.99	1.0	0.99	1.0

Table 23: Constituency Tree Normalized Subtree Kernel scores within each subcase for subcases with only 1 template. We replace POS tags and leaf nodes in the tree with a dummy token x to compare the similarity in the overall tree structures. We expect the parsed trees within each subcase with only 1 template to be highly similar (Normalized Subtree Kernel=1). Scores below 1 indicate deviation from the template.

NLI evaluates the capabilities of natural language systems at the sentence level, as opposed to word level for the task of Semantic Role Labeling. Moreover, our results, in conjunction with results by Glavaš and Vulić (2021), indicate that syntactic information does not play a prominent role in general-purpose NLI datasets, and the gains made from using more correct parses may therefore be minimal.

7 Conclusion and Outlook

In this thesis, we have enhanced the *BERT-base-uncased* model with linguistically informed syntactic structures and evaluated the effects of doing so on various generic large-scale NLI benchmarks, as well as NLI evaluation sets designed to test specific linguistic capabilities of language models. In summary, our results show that for general-purpose NLI datasets, enhancing BERT with syntax via GCNs may overfit the model on spurious correlations such as hypothesis-only biases, as well as other aspects of natural language not related to syntactic knowledge such as common sense and world knowledge. Nevertheless, when we curate the datasets of a vast majority of hypothesis-only biases, we find that our syntax models are better able to pick up on the syntactic signals in the training data, and our syntax models are able to outperform the baseline on the evaluation sets of SNLI. Likewise, we have found that enhancing BERT with syntactic structures can be beneficial on datasets rich in syntactic information, where the underlying syntactic structure is important for the inference task, and models enhanced with syntax learn more useful representations from these datasets. In addition, our results indicate that syntax helps the models adapt towards new distributions when data is scarce, and constituency structures aid in learning useful representations for passive sentences and the linguistic phenomena of monotonicity reasoning when provided sufficient signal from the training data.

In the wider context of NLP, the role of imbuing language models with syntax for NLU is the subject of ongoing research and debate (Kulmizev and Nivre, 2022). We

have provided empirical data for the effects of enhancing the BERT-base model with syntactic structures for the task of NLI using a GCN-based approach. Nonetheless, alternative methodologies for incorporating linguistic structures into language models exist. For example, Chen (2021) show that incorporating dependency structures into tree-LSTM models can improve performance for the monotonicity reasoning task over the base BERT model. Likewise, the methodology proposed by Chen et al. (2021), which synergistically integrates the robustness of neural networks with expert linguistic knowledge within a hybrid system, has exhibited exceptional performance for the task of NLI, particularly for monotonicity reasoning. Yet, their results are limited to small-scale datasets, leaving the exploration of such methods on generic, large-scale datasets as intriguing avenues to explore.

Despite the existence of alternative methodologies, our findings and discussions open up several possibilities for follow-up research closely connected to our own[29]. Future work may explore modifications to the model architecture to increase performance for the task of NLI. This could include restructuring the order of the GCN components in the DC model by first passing the BERT embeddings through the Dep component and subsequently through the Con component. While our model architecture is optimal for Semantic Role Labeling (Fei et al., 2021), its effectiveness in NLI is yet to be confirmed. Moreover, altering the Con and Dep components to accommodate scalability might be another area of consideration. For the Con component, including additional GCN layers between the Span-boundary Bridging and Span-boundary Inverse Bridging operations can be an option. For the Dep component, the additional layers may necessitate the removal of the dependency labels, as including them at every layer may overparameterize the network. Contrasting findings with He et al. (2020) on the HANS dataset underscore the need for architectural scaling by deepening the GCN components. In addition, it would be of interest to examine how our results translate to LLM-based models which have undergone more robust pre-training regimes or are scaled in size, such as RoBERTa or DeBERTa. Besides architectural modifications, future work may include investigating to what extent our syntax-enhanced models are word order insensitive, in line with work by Sinha et al. (2021) and Pham et al. (2021). Lastly, considering our results on the ANLI evaluation set, which contains lengthy, multi-sentence premises, future work might investigate the effects of enhancing BERT with syntactic structures on varying context lengths. This is particularly important for document-level NLI, and follow-up research in this direction may provide valuable insights into the broader landscape of NLP.

In conclusion to this thesis, we have investigated the effects of enhancing a BERT-based model with dependency and constituency structures via GCNs for the task of NLI. Nevertheless, the process of incorporating syntactic knowledge into language models, as well as the importance of linguistic syntax itself, continue to be pivotal yet unresolved questions in the task of Natural Language Inference.

[29] Our code is publicly available at https://github.com/lucalin17081994/Syntax-Enhanced-Bert.

8 Acknowledgements

I would like to extend my sincere appreciation to my daily supervisor, Lasha Abzianidze, for his patience, his scientific guidance, and his feedback throughout the writing of this thesis.

I would also like to thank the ITS and HPC departments at Utrecht University for their assistance in providing us with access to the computational resources necessary for the completion of this thesis. We thank SURF (www.surf.nl) for the support in using the National Supercomputer Snellius.

References

Xuesong Zhai, Xiaoyan Chu, Ching Sing Chai, Morris Siu Yung Jong, Andreja Istenic, Michael Spector, Jia-Bao Liu, Jing Yuan, and Yan Li. A review of artificial intelligence (ai) in education from 2010 to 2020. *Complexity*, 2021:1–18, 2021.

Sida Peng, Eirini Kalliamvakou, Peter Cihon, and Mert Demirer. The impact of ai on developer productivity: Evidence from github copilot, 2023.

Ido Dagan and Oren Glickman. Probabilistic textual entailment: Generic applied modeling of language variability. 2004.

Moritz Laurer, W v Atteveldt, Andreu Casas, and Kasper Welbers. Less annotating, more classifying–addressing the data scarcity issue of supervised machine learning with deep transfer learning and bert-nli, 2022.

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 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL https://aclanthology.org/D15-1075.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://aclanthology.org/N18-1101.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018. URL https://arxiv.org/abs/1810.04805.

Noam Chomsky. Syntactic structures. the hague: Mouton.. 1965. aspects of the theory of syntax. *Cambridge, Mass.: MIT Press.(1981) Lectures on Government and Binding, Dordrecht: Foris.(1982) Some Concepts and Consequences of the Theory of Government and Binding. LI Monographs*, 6:1–52, 1957.

Igor Aleksandrovic Mel'cuk et al. *Dependency syntax: theory and practice*. SUNY press, 1988.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016. URL http://arxiv.org/abs/1609.02907.

Zeming Chen, Qiyue Gao, and Lawrence S. Moss. NeuralLog: Natural language inference with joint neural and logical reasoning. In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 78–88, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.starsem-1.7. URL https://aclanthology.org/2021.starsem-1.7.

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://aclanthology.org/P19-1334.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. Can neural networks understand monotonicity reasoning? In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 31–40, Florence, Italy, August 2019a. Association for Computational Linguistics. doi: 10.18653/v1/W19-4804. URL https://aclanthology.org/W19-4804.

Jiangang Bai, Yujing Wang, Yiren Chen, Yaming Yang, Jing Bai, Jing Yu, and Yunhai Tong. Syntax-BERT: Improving pre-trained transformers with syntax trees. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3011–3020, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.262. URL https://aclanthology.org/2021.eacl-main.262.

Junru Zhou, Zhuosheng Zhang, Hai Zhao, and Shuailiang Zhang. LIMIT-BERT : Linguistics informed multi-task BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4450–4461, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.399. URL https://aclanthology.org/2020.findings-emnlp.399.

Diego Marcheggiani and Ivan Titov. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1159. URL https://aclanthology.org/D17-1159.

Qi He, Han Wang, and Yue Zhang. Enhancing generalization in natural language inference by syntax. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4973–4978, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.447. URL https://aclanthology.org/2020.findings-emnlp.447.

Hao Fei, Shengqiong Wu, Yafeng Ren, Fei Li, and Donghong Ji. Better combine them together! integrating syntactic constituency and dependency representations for semantic role labeling. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 549–559, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.49. URL https://aclanthology.org/2021.findings-acl.49.

Zeming Chen. Attentive tree-structured network for monotonicity reasoning. In *Proceedings of the 1st and 2nd Workshops on Natural Logic Meets Machine Learning (NALOMA)*, pages 12–21, Groningen, the Netherlands (online), June 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.naloma-1.3.

Julia Rozanova, Deborah Ferreira, Mokanarangan Thayaparan, Marco Valentino, and Andre Freitas. Decomposing natural logic inferences for neural NLI. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 394–403, Abu Dhabi, United Arab Emirates (Hybrid), December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.blackboxnlp-1.33.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. SemEval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 1–8, Dublin, Ireland, August 2014. Association for Computational Linguistics. doi: 10.3115/v1/S14-2001. URL https://aclanthology.org/S14-2001.

Stefan Evert. Distributional semantic models. In *NAACL HLT 2010 Tutorial Abstracts*, pages 15–18, Los Angeles, California, June 2010. Association for Computational Linguistics. URL https://aclanthology.org/N10-4006.

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. Von 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.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. URL https://arxiv.org/abs/2010.11929.

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lucic, and Cordelia Schmid. Vivit: A video vision transformer. *CoRR*, abs/2103.15691, 2021. URL https://arxiv.org/abs/2103.15691.

Fang Wu, Dragomir Radev, and Stan Z. Li. Molformer: Motif-based transformer on 3d heterogeneous molecular graphs, 2023.

Linhao Dong, Shuang Xu, and Bo Xu. Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5884–5888, 2018.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997. doi: 10.1162/neco.1997.9.8.1735.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013. URL https://arxiv.org/abs/1301.3781.

Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL https://aclanthology.org/D14-1162.

Dumitru Erhan, Pierre-Antoine Manzagol, Yoshua Bengio, Samy Bengio, and Pascal Vincent. The difficulty of training deep architectures and the effect of unsupervised pre-training. In David van Dyk and Max Welling, editors, *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 153–160, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 16–18 Apr 2009. PMLR. URL https://proceedings.mlr.press/v5/erhan09a.html.

Dumitru Erhan, Aaron Courville, Yoshua Bengio, and Pascal Vincent. Why does unsupervised pre-training help deep learning? In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 201–208, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL https://proceedings.mlr.press/v9/erhan10a.html.

Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, Avi Schwarzschild, Andrew Gordon Wilson, Jonas Geiping, Quentin Garrido, Pierre Fernandez, Amir Bar, Hamed Pirsiavash, Yann LeCun, and Micah Goldblum. A cookbook of self-supervised learning, 2023.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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. *CoRR*, abs/2005.14165, 2020. URL https://arxiv.org/abs/2005.14165.

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://aclanthology.org/P17-1152.

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. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

OpenAI. Gpt-4 technical report, 2023.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108, 2019. URL http://arxiv.org/abs/1910.01108.

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 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2017. URL https://aclanthology.org/N18-2017.

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://aclanthology.org/S18-2023.

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. URL https://www.aclweb.org/anthology/D17-1070.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomás Mikolov. Bag of tricks for efficient text classification. *CoRR*, abs/1607.01759, 2016. URL http://arxiv.org/abs/1607.01759.

Mor Geva, Yoav Goldberg, and Jonathan Berant. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. 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 1161–1166, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1107. URL https://aclanthology.org/D19-1107.

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 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.569. URL https://aclanthology.org/2021.acl-long.569.

Thang 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? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1145–1160, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.98. URL https://aclanthology.org/2021.findings-acl.98.

Goran Glavaš and Ivan Vulić. Is supervised syntactic parsing beneficial for language understanding tasks? an empirical investigation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3090–3104, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.270. URL https://aclanthology.org/2021.eacl-main.270.

Chen Xu, Jun Xu, Zhenhua Dong, and Ji-Rong Wen. Semantic sentence matching via interacting syntax graphs. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 938–949, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics. URL https://aclanthology.org/2022.coling-1.78.

Lingfei Wu, Yu Chen, Kai Shen, Xiaojie Guo, Hanning Gao, Shucheng Li, Jian Pei, and Bo Long. Graph neural networks for natural language processing: A survey. *CoRR*, abs/2106.06090, 2021. URL https://arxiv.org/abs/2106.06090.

Petar Veličković. Everything is connected: Graph neural networks, 2023.

Ziyang Luo. Have attention heads in BERT learned constituency grammar? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 8–15, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-srw.2. URL https://aclanthology.org/2021.eacl-srw.2.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SJzSgnRcKX.

Yichu Zhou and Vivek Srikumar. A closer look at how fine-tuning changes BERT. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1046–1061, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.75. URL https://aclanthology.org/2022.acl-long.75.

Haoyue Shi, Hao Zhou, Jiaze Chen, and Lei Li. On tree-based neural sentence modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4631–4641, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1492. URL https://aclanthology.org/D18-1492.

Changlong Yu, Tianyi Xiao, Lingpeng Kong, Yangqiu Song, and Wilfred Ng. An empirical revisiting of linguistic knowledge fusion in language understanding tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10064–10070, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.emnlp-main.684.

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://aclanthology.org/W18-5446.

Junghyun Min, R. Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. Syntactic data augmentation increases robustness to inference heuristics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2339–2352, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.212. URL https://aclanthology.org/2020.acl-main.212.

Yuxiang Wu, Matt Gardner, Pontus Stenetorp, and Pradeep Dasigi. Generating data to mitigate spurious correlations in natural language inference datasets. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2660–2676, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.190. URL https://aclanthology.org/2022.acl-long.190.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. HELP: A dataset for identifying shortcomings of neural models in monotonicity reasoning. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 250–255, Minneapolis, Minnesota, June 2019b. Association for Computational Linguistics. doi: 10.18653/v1/S19-1027. URL https://aclanthology.org/S19-1027.

Diego Marcheggiani and Ivan Titov. Graph convolutions over constituent trees for syntax-aware semantic role labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3915–3928, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.322. URL https://aclanthology.org/2020.emnlp-main.322.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *CoRR*, abs/1505.00387, 2015. URL http://arxiv.org/abs/1505.00387.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A python natural language processing toolkit for many human languages. *CoRR*, abs/2003.07082, 2020. URL https://arxiv.org/abs/2003.07082.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2017. URL https://arxiv.org/abs/1711.05101.

Natalie Schluter and Daniel Varab. When data permutations are pathological: the case of neural natural language inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4935–4939, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1534. URL https://aclanthology.org/D18-1534.

Alessandro Moschitti. Making tree kernels practical for natural language learning. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 113–120, Trento, Italy, April 2006. Association for Computational Linguistics. URL https://aclanthology.org/E06-1015.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training, 2021.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. doi: 10.1162/tacl_a_00166. URL https://aclanthology.org/Q14-1006.

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. The Parallel Meaning Bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 242–247, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://aclanthology.org/E17-2039.

Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899, 2013.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics. URL https://aclanthology.org/S12-1051.

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://aclanthology.org/2020.acl-main.441.

Yixin Nie, Haonan Chen, and Mohit Bansal. Combining fact extraction and verification with neural semantic matching networks. *CoRR*, abs/1811.07039, 2018. URL http://arxiv.org/abs/1811.07039.

Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. WANLI: Worker and AI collaboration for natural language inference dataset creation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6826–6847, Abu Dhabi, United Arab Emirates, December 2022. Association for Compu-

tational Linguistics. URL https://aclanthology.org/2022.findings-emnlp.508.

Alicia Parrish, William Huang, Omar Agha, Soo-Hwan Lee, Nikita Nangia, Alexia Warstadt, Karmanya Aggarwal, Emily Allaway, Tal Linzen, and Samuel R. Bowman. Does putting a linguist in the loop improve NLU data collection? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4886–4901, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.421. URL https://aclanthology.org/2021.findings-emnlp.421.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259.

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 the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1098. URL https://aclanthology.org/N16-1098.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. The goldilocks principle: Reading children's books with explicit memory representations. *arXiv preprint arXiv:1511.02301*, 2015.

Aikaterini-Lida Kalouli, Livy Real, and Valeria de Paiva. Textual inference: getting logic from humans. In *IWCS 2017 — 12th International Conference on Computational Semantics — Short papers*, 2017. URL https://aclanthology.org/W17-6915.

Adina Williams, Tristan Thrush, and Douwe Kiela. ANLIzing the adversarial natural language inference dataset. In *Proceedings of the Society for Computation in Linguistics 2022*, pages 23–54, online, February 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.scil-1.3.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts, 2023.

Yangfan Lei, Yue Hu, Xiangpeng Wei, Luxi Xing, and Quanchao Liu. Syntax-aware sentence matching with graph convolutional networks. In Christos Douligeris,

Dimitris Karagiannis, and Dimitris Apostolou, editors, *Knowledge Science, Engineering and Management*, pages 353–364, Cham, 2019. Springer International Publishing. ISBN 978-3-030-29563-9.

Michael McCloskey and Neal J. Cohen. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. volume 24 of *Psychology of Learning and Motivation*, pages 109–165. Academic Press, 1989. doi: https://doi.org/10.1016/S0079-7421(08)60536-8. URL https://www.sciencedirect.com/science/article/pii/S0079742108605368. ISSN: 0079-7421.

Artur Kulmizev and Joakim Nivre. Schrödinger's tree—on syntax and neural language models. *Frontiers in Artificial Intelligence*, 5:796788, 2022.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention, 2021.

Devendra Sachan, Yuhao Zhang, Peng Qi, and William L. Hamilton. Do syntax trees help pre-trained transformers extract information? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2647–2661, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.228. URL https://aclanthology.org/2021.eacl-main.228.

A Appendix

A.1 Dependency Label Mappings

In this subsection, we provide an overview of the mappings used for experiments investigating the effects of clustering similar and infrequent dependency labels. Fine-grained condition is the original Stanza parsed dictionary used by the Dep. Table 24 shows the mappings from fine-grained to mid-grained (Mid) and coarse-grained conditions (Coarse). We split the table for readability purposes.

Dep Label	Mid	Coarse	Dep Label	Mid	Coarse
det	det	det	expl	expl	null
case	case	case	acl:relcl	acl	mod
punct	punct	null	appos	appos	mod
root	root	root	parataxis	parataxis	mod
nsubj	nsubj	subj	obl:agent	obl:agent	obl
obl	obl	obl	ccomp	ccomp	obj
amod	amod	mod	fixed	fixed	mod
obj	obj	obj	flat	fixed	mod
nmod	nmod	mod	obl:tmod	obl	obl
aux	aux	null	obl:npmod	obl	obl
compound	compound	mod	iobj	iobj	obj
acl	acl	mod	cc:preconj	cc	null
conj	conj	mod	det:predet	det	det
cc	cc	null	csubj	csubj	subj
nummod	nummod	mod	discourse	punct	null
mark	mark	mark	dep	conj	mod
nmod:poss	nmod	mod	nmod:npmod	nmod	mod
advmod	advmod	mod	list	conj	mod
advcl	advcl	mod	vocative	mod	null
cop	cop	null	nmod:tmod	nmod	mod
xcomp	xcomp	xcomp	orphan	obj	obj
compound:prt	compound	mod	dislocated	conj	mod
aux:pass	aux	null	reparandum	conj	mod
nsubj:pass	nsubj:pass	subj:pass	goeswith	fixed	mod
			csubj:pass	nsubj:pass	nsubj:pass

Table 24: Dependency label mappings for the experiments on dependency label granularity (RQ1.a). The first, second, and third columns represent the original dependency label, and their respective mappings for the mid-grained (Mid) and coarse-grained (Coarse) conditions. The table has been split for readability.

A.2 Additional Fine-grained Results

	Entailed			Non-entailed			Average		
Model	lex	sub	const	lex	sub	const	E	NE	all
BERT_S	88.42	97.56	98.88	**58.18**	11.14	5.52	94.95	24.95	59.95
Con_S	**98.18**	99.00	99.78	50.18	4.24	2.48	**98.99**	18.97	58.98
Dep_S	87.10	96.68	95.3	55.18	**18.56**	**12.04**	93.03	**28.59**	**60.81**
DC_S	96.88	**99.74**	**99.98**	45.20	4.78	1.82	98.87	17.27	58.07
BERT_M	90.76	98.32	98.86	24.04	4.74	20.16	95.98	16.31	56.15
Con_M	94.92	99.06	99.16	25.12	3.18	18.74	97.71	15.68	56.70
Dep_M	90.8	97.58	98.18	21.4	5.64	18.9	95.52	15.31	55.42
DC_M	91.42	98.94	99.58	27.86	4.84	13.8	96.65	15.50	56.07
BERT_M (He et al. (2020))	**97.5**	**99.8**	**99.8**	33.4	4.3	12.9	99.0	16.87	57.9
BERT+SGCN_M (He et al. (2020))	94.6	98.9	99.0	49.2	8.3	13.0	97.5	23.5	60.5
CAGCN_M (He et al. (2020))	94.9	99.5	98.9	**64.0**	**8.8**	**14.6**	97.77	29.13	**63.5**

Table 25: Fine-grained performances on HANS for models trained on SNLI (S) or MNLI (M) and results provided by He et al. (2020). Lex, sub, and const are the lexical-overlap, subsequence, and constituent heuristics. E represents entailment cases, whereas NE represents non-entailed cases.

	Entailed			Non-entailed			Average		
Model	lex	sub	const	lex	sub	const	E	NE	all
BERT_S	99.94	98.58	98.98	0.06	0.4	1.54	99.17	0.67	49.92
Con_S	99.54	99.74	98.4	1	0.42	3.36	99.23	**1.59**	**50.41**
Dep_S	99.96	99.94	99.98	0	0.12	0.1	**99.96**	0.07	50.02
DC_S	99.86	99.98	99.7	0.08	0.16	0.44	99.85	0.23	50.04
BERT_M	98.88	97.8	97.14	3.42	5.26	3.6	97.94	4.09	51.02
Con_M	99.1	96.7	98.66	5.44	4.08	2.86	98.15	4.13	51.14
Dep_M	98.04	96	96.74	9.36	4.68	6.06	96.93	**6.7**	**51.81**
DC_M	99.32	96.76	99.06	4.22	2.92	2.76	**98.38**	3.3	50.84

Table 26: Fine-grained performances on HANS for models trained on SNLI-train-hard (S) or MNLI-train-hard (M). Lex, sub, and const are the lexical-overlap, subsequence, and constituent heuristics. E represents entailment cases, whereas NE represents non-entailed cases.

Model	Up	Down	Non	All
BERT_S	**80.05**	31.50	50.34	48.94
Con_S	79.12	41.74	**57.88**	**55.26**
Dep_S	76.26	36.06	51.71	50.50
DC_S	74.12	**44.56**	51.71	54.94
BERT_M	79.56	29.36	48.29	47.36
Con_M	80.38	29.7859	51.37	48.07
Dep_M	**80.77**	27.77	48.63	46.82
DC_M	79.07	**32.14**	**51.71**	**49.07**

Table 27: Performance on MED test set for models trained on SNLI-train-hard (S) or MNLI-train-hard (M).

	Entailed			Non-Entailed			Average		
Model	lex	subseq	const	lex	subseq	const	ent	non-ent	average
BERT_{SICK}	83.50	96.84	92.52	62.52	25.92	16.32	90.95	34.92	62.94
Con_{SICK}	**89.38**	94.52	95.40	71.08	23.52	16.68	93.10	37.09	65.10
Dep_{SICK}	71.70	93.64	82.84	**72.18**	**42.78**	**30.32**	82.73	**48.43**	**65.58**
DC_{SICK}	88.32	**97.52**	96.42	61.14	27.48	14.36	**94.09**	34.33	64.21

Table 28: Performance on HANS validation set for models trained on SNLI and further fine-tuned on SICK.

A.3 Constituency Tree Similarity Scores Stanza vs Original HANS

	Original HANS		Stanza	
Subcase	Premise	Hypothesis	Premise	Hypothesis
ln_subject/object_swap	1.0	1.0	0.95	0.95
ln_preposition	0.58	1.0	0.59	0.93
ln_relative_clause	0.65	1.0	0.59	0.85
ln_passive	1.0	1.0	1.0	0.94
ln_conjunction	0.79	1.0	0.75	0.94
le_relative_clause	0.65	1.0	0.59	0.95
le_around_prepositional_phrase	0.71	1.0	0.71	0.95
le_around_relative_clause	0.67	1.0	0.62	0.94
le_conjunction	0.79	1.0	0.76	0.94
le_passive	1.0	1.0	1.0	0.94
sn_NP/S	0.69	1.0	0.66	0.79
sn_PP_on_subject	0.61	0.78	0.64	0.75
sn_relative_clause_on_subject	0.89	0.77	0.73	0.75
sn_past_participle	0.78	1.0	0.63	0.96
sn_NP/Z	0.86	1.0	0.63	1.0
se_conjunction	0.72	0.83	0.70	0.80
se_adjective	0.79	0.77	0.77	0.75
se_understood_object	0.99	1.0	0.99	1.0
se_relative_clause_on_obj	0.60	1.0	0.57	0.93
se_PP_on_obj	0.64	1.0	0.61	0.95
cn_embedded_under_if	0.65	0.77	0.62	0.75
cn_after_if_clause	0.74	0.76	0.69	0.74
cn_embedded_under_verb	0.69	0.77	0.67	0.75
cn_disjunction	0.76	0.77	0.70	0.75
cn_adverb	0.78	0.78	0.79	0.75
ce_embedded_under_since	0.74	0.77	0.69	0.75
ce_after_since_clause	0.69	0.77	0.65	0.74
ce_embedded_under_verb	0.69	0.77	0.66	0.75
ce_conjunction	0.77	0.76	0.73	0.74
ce_adverb	0.65	0.78	0.67	0.75

Table 29: Full table of constituency Tree Normalized Subtree Kernel scores within each subcase in HANS. Trees within each subcase with only 1 template should be highly similar because they are generated through the same template (ST=1). Most subcases in HANS have more than 1 template.

A.4 ANLI Development Set Results Per Category

N	Tag	BERT	Con	Dep	DC-GCN
893	Tricky	42.55	40.87	38.97	42.22
199	Exhaustification	35.68	32.66	30.65	35.68
1327	Basic	43.56	43.93	43.71	44.99
173	Coordination	49.13	49.71	47.4	49.71
678	Lexical	44.69	46.76	45.28	47.05
576	Similar	45.83	46.88	46.01	48.61
1036	Numerical	47.2	48.36	47.1	49.42
948	Cardinal	48.1	48.84	48.21	50.42
602	Dates	50.0	52.33	49.17	52.49
82	Nominal	45.12	45.12	47.56	48.78
1977	Reasoning	48.0	46.99	47.34	47.7
1030	Plausibility	53.98	49.81	53.01	51.07
768	Likely	51.43	47.66	51.43	50.13
868	Reference	46.2	45.62	44.93	46.43
691	Coreference	45.44	45.3	44.86	46.6
343	Negation	44.31	44.31	44.61	44.31
129	Translation	41.09	44.96	39.53	40.31
43	Family	34.88	34.88	34.88	34.88
68	CauseEffect	32.35	38.24	35.29	35.29
452	Imperfection	45.13	46.02	43.58	46.02
189	Spelling	44.97	46.56	48.15	48.68
143	Wordplay	59.44	55.94	55.94	55.94
260	Names	46.15	46.92	44.62	46.92
84	Error	35.71	45.24	39.29	34.52
769	Facts	39.66	42.65	40.05	42.39
162	Ordinal	43.21	46.3	41.98	43.83
261	Unlikely	61.69	56.32	57.85	54.02
39	NonNative	46.15	43.59	43.59	43.59
74	Counting	43.24	50.0	45.95	50.0
277	Containment	50.18	48.01	47.29	46.21
137	Times	53.28	48.91	48.91	46.72
225	Debatable	48.44	46.67	46.67	50.67
188	ComparativeSuperlative	45.21	40.96	45.21	44.15
155	Pragmatic	37.42	34.19	35.48	40.65
337	Syntactic	41.25	40.06	35.31	40.06
164	Age	52.44	55.49	49.39	51.22
106	Dissimilar	37.74	46.23	39.62	39.62
125	Location	48.0	48.8	46.4	46.4
60	0	46.67	46.67	46.67	51.67
159	Ambiguity	45.28	42.14	39.62	42.77
22	Modus	27.27	27.27	31.82	40.91
15	Parts	40.0	33.33	40.0	40.0
46	Idiom	23.91	30.43	30.43	36.96
1	Quality	100.0	100.0	100.0	100.0
66	EventCoref	46.97	51.52	51.52	46.97

Table 30: Fine-grained performances for each model trained on the concatenation of MNLI, ANLI, FeverNLI, LingNLI, and WaNLI on the annotated development set of ANLI for each category.

	BERT	Con	Dep	DC
fiction	0.74	0.74	0.73	0.75
government	0.78	0.78	0.76	0.77
slate	0.70	0.71	0.70	0.71
telephone	0.74	0.73	0.74	0.73
travel	0.75	0.75	0.74	0.75

Table 31: Performance by genre for each model on MNLI-matched evaluation set when trained on MNLI-train-hard.

	BERT	Con	Dep	DC
facetoface	0.75	0.76	0.74	0.75
letters	0.77	0.77	0.77	0.77
nineeleven	0.75	0.76	0.76	0.76
oup	0.75	0.75	0.75	0.76
verbatim	0.72	0.73	0.71	0.73

Table 32: Performance by genre of each model on MNLI-mismatched evaluation set when trained on MNLI-train-hard.