Improving Natural Language Inference with Syntactic Word Representations

Deric Pang

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Master of Science

University of Washington

2019

Reading Committee:

Noah A. Smith, Chair

Richard Anderson

Jenifer Hiigli

Program Authorized to Offer Degree:
Paul G. Allen School of Computer Science & Engineering

University of Washington

Abstract

Improving Natural Language Inference
with Syntactic Word Representations

Deric Pang

Chair of the Supervisory Committee:
Professor Noah A. Smith
Paul G. Allen School of Computer Science & Engineering

We introduce a novel approach to incorporate syntax into natural language inference (NLI) models. Our method uses contextual token-level vector representations from a pre-trained dependency parser. Like other contextual embedders, our method is broadly applicable to any neural model. We experiment with four strong NLI models (decomposable attention model, ESIM, BERT, and MT-DNN), and show consistent benefit to accuracy across three NLI benchmarks.

TABLE OF CONTENTS

ACKNOWLEDGMENTS

I would like to thank my advisor, Noah Smith, not only for his technical guidance, but also for his unwavering encouragement which has helped me overcome many obstacles in my research. Without his efforts, this work would not have been possible.

I would also like to thank my advisor during my undergraduate years, Michael Ernst, for his careful mentorship. His thoughtful advice has helped me through many difficult decisions.

Thanks also to René Just, Lucy Lin, and Victoria Lin for their mentorship. René gave me my first opportunity to conduct research when I was only a freshman with six months of programming experience, and I am grateful for the many hours he spent patiently answering my questions. Lucy has been a critical part of this work and has been vital to my growth as a researcher, and Victoria has given me valuable advice and feedback for both my research and career.

Thanks to all my friends and colleagues in the ARK research group, the Allen School, and the University of Washington. Special thanks to Jenifer Hiigli who has been an amazing academic advisor.

Finally, thanks to my family for their unconditional love and support.

Chapter 1

INTRODUCTION

We consider natural language inference (NLI) tasks in which the semantic relationship between two sentences is classified as entailment, contradiction, or neither. Our focus is on the use of syntactic representations of the sentences, given (1) longstanding linguistic theory that posits a close relationship between syntax and semantics (e.g., Montague, 1970; Steedman and Baldridge, 2011) and (2) major advances in syntactic parsing accuracy (e.g., Kiperwasser and Goldberg, 2016; Dozat and Manning, 2017).

Building on the recent success of transferring contextual word representations learned by large-data language modeling or translation to other tasks (McCann et al., 2017; Peters et al., 2018; Radford et al., 2018; Devlin et al., 2019), we introduce two general methods for transferring word representations from a *parser* to an NLI model (§3) — through input into a model's final feedforward (classification) layer, or through a model's attention mechanism.

We see widespread gains when applying these methods to four strong NLI models (decomposable attention model, ESIM, BERT, and MT-DNN) on the SNLI, MNLI and SciTail datasets (§4). We also provide analysis for how including syntactic representations from a parser is helpful. In particular, we probe our models with the Heuristic Analysis for NLI Systems (HANS) dataset and show that when our methods improve NLI test accuracy, performance on this evaluation dataset improves as well (§5).

Chapter 2
BACKGROUND

In this chapter, we will introduce the natural language inference task and the four baseline models we experiment with: the decomposable attention model, ESIM, BERT, and MT-DNN.

2.1 Task Setup

Natural language inference (NLI) is the task of characterizing entailment and contradiction relationships between texts. In general, NLI tasks are formulated as predicting the relationship between a pair of sequences—a premise and a hypothesis. An NLI model should predict whether the hypothesis is entailed by the premise, contradicts the premise, or is neutral to the premise. An example of two sentences (from the SciTail dataset) exhibiting the *entailment* relationship is shown below.

(1) $\boldsymbol{p}$: Beats are the periodic and repeating fluctuations heard in the intensity of a sound when two sound waves of very similar frequencies interfere with one another.

$\boldsymbol{h}$: When waves of two different frequencies interfere, beating occurs.

2.2 Basic NLI Model Structure

Let the premise be represented as a sequence of token embeddings, $\langle \boldsymbol{p}_1, \ldots, \boldsymbol{p}_{\ell_p} \rangle$, and likewise for the hypothesis, $\langle \boldsymbol{h}_1, \ldots, \boldsymbol{h}_{\ell_h} \rangle$. An NLI model will take these sequences as input and apply some series of transformations to create an output vector $\boldsymbol{e}$. From there, a final feedforward layer (H) is applied to $\boldsymbol{e}$ to compute the final prediction:

$$\hat{\boldsymbol{y}} = H(\boldsymbol{e}). \tag{2.1}$$

2.3 Decomposable Attention Model

The decomposable attention model (DA; Parikh et al., 2016) is a simple and highly parallelizable model comprised of attend, compare, and aggregate steps.

Attend. Compute unnormalized attention weights e_{ij} with a feed-forward neural network F:

$$e_{ij} := F(p_i)^\mathsf{T} F(h_j). \tag{2.2}$$

Compute the aligned subphrases $\boldsymbol{H}i$ and $\boldsymbol{P}_j$:

$$\begin{aligned} \boldsymbol{H}_i &:= \sum_{j=1}^{\ell_h} \frac{\exp\left(e_{ij}\right)}{\sum_{k=1}^{\ell_h} \exp\left(e_{ik}\right)} h_j \,, \\ \boldsymbol{P}_j &:= \sum_{i=1}^{\ell_p} \frac{\exp\left(e_{ij}\right)}{\sum_{k=1}^{\ell_h} \exp\left(e_{ik}\right)} p_i \end{aligned} \tag{2.3}$$

where $\boldsymbol{H}_i$ is the weighted sum over $\boldsymbol{h}$ that aligns to p_i and vice versa for $\boldsymbol{P}_j$.

Compare. Compare each $(p_i, \boldsymbol{H}_i)$ and $(h_j, \boldsymbol{P}_j)$ pairs with another feed-forward neural network G:

$$\begin{aligned} \boldsymbol{v}_i^p &:= G([p_i, \boldsymbol{H}_i]) \quad \forall i \in [1, \ldots, \ell_p] \,, \\ \boldsymbol{v}_j^h &:= G([h_j, \boldsymbol{P}_j]) \quad \forall j \in [1, \ldots, \ell_h] \,. \end{aligned} \tag{2.4}$$

Aggregate. Aggregate each set of comparison vectors:

$$\boldsymbol{v}^p = \sum_{i=1}^{\ell_p} \boldsymbol{v}_i^p \qquad , \qquad \boldsymbol{v}^h = \sum_{j=1}^{\ell_h} \boldsymbol{v}_j^h \,. \tag{2.5}$$

and make a prediction with a final feed-forward layer H:

$$\hat{\boldsymbol{y}} = H([\boldsymbol{v}^p, \boldsymbol{v}^h]). \tag{2.6}$$

There is a variation of DA which uses intra-sentence attention to take word order into account, but most evaluations (including ours) use the original DA model for comparison.

2.4 ESIM

The Enhanced Sequence Inference Model (ESIM; Chen et al., 2017) uses bidirectional LSTMs with attention. The ESIM model is comprised of (1) input encoding, (2) local inference modeling, and (3) inference composition steps. Steps (1) and (3) are performed with bidirectional LSTMs, and step (2) is computed with a dot-product as described by equation 2.2. We use the sequential variant of ESIM without syntactic parsing information in tree LSTMs.

2.5 BERT

Bidirectional Encoder Representations from Transformers (BERT; Devlin et al., 2019) is a transformer model pretrained on massive amounts of text. It can be fine-tuned to a specific dataset or task. In order to use BERT for classification, we train an additional linear layer on top of the original BERT model.

We refer the reader to the original BERT and transformer papers for more details (Devlin et al., 2019; Vaswani et al., 2017).

2.6 MT-DNN

The Multi-Task Deep Neural Network for Natural Language Understanding (MT-DNN; Liu et al., 2019b) is a carefully fine-tuned BERT model multi-tasked on the nine GLUE tasks (Wang et al., 2019). Like BERT, it can be fine-tuned to a specific dataset or task. At publication time, MT-DNN was state of the art on SNLI, SciTail, and GLUE; it is still state of the art on MNLI.

Chapter 3

MODEL

We propose a method of extracting syntactic word representations from a pretrained parser and introduce two general approaches to incorporate syntactic word representations in NLI models.

3.1 Extracting Syntactic Word Representations

Given a neural syntax parser with an encoder (e.g., LSTM or transformer) that encodes a representation of the input sequence from which a parse is later computed, we can obtain contextual token-level vector representations of the premise and hypothesis, $\langle \boldsymbol{s}^p_1, \ldots, \boldsymbol{s}^p_{\ell_p} \rangle$ and $\langle \boldsymbol{s}^h_1, \ldots, \boldsymbol{s}^h_{\ell_h} \rangle$, which we call "syntactic word representations" (SWRs). In our method, $\boldsymbol{s}^p$ and $\boldsymbol{s}^h$ are the hidden states of the encoder when (separately) parsing the premise and hypothesis.

3.2 Late Fusion of SWRs

The simplest way to incorporate the extracted SWRs is to attach them to $\boldsymbol{e}$, the input to the final feedforward layer. More precisely, we concatenate the final contextual representations from the parser ($\boldsymbol{s}^p_{\ell_p}$, $\boldsymbol{s}^h_{\ell_h}$) to $\boldsymbol{e}$. The prediction step in Equation 2.1 then becomes:

$$\hat{\boldsymbol{y}} = H([\boldsymbol{e}, \boldsymbol{s}^p_{\ell_p}, \boldsymbol{s}^h_{\ell_h}]) \tag{3.1}$$

This approach is very general, since many neural models use a final feedforward classification layer, and adds a minimal number of parameters. We will refer to this variant of our method as Late Fusion (`+LF`; Fig. 3.1, left).

3.3 Using SWRs to Attend

A natural place to include SWRs in models that use attention is in conjunction with the attention weights; that is, perhaps syntax would be helpful for guiding soft alignments between subphrases. Calculating attention is often formulated as the dot product between two sequence representations. In NLI, this will be an encoded representation of the premise and hypothesis $(\bar{\boldsymbol{p}}, \bar{\boldsymbol{h}})$. Attention between the ith and jth tokens in the premise and hypothesis is calculated with:

$$a_{ij} := \bar{\boldsymbol{p}}_i^\top \bar{\boldsymbol{h}}_j \tag{3.2}$$

and used downstream to ultimately compute $\boldsymbol{e}$.

To augment the attention calculation with SWRs, we modify Equation 3.2 to be:

$$a_{ij} := [\bar{\boldsymbol{p}}_i, \boldsymbol{s}_i^p]^\top [\bar{\boldsymbol{h}}_j, \boldsymbol{s}_j^h]. \tag{3.3}$$

Although the model will not learn any weights directly on the SWRs, this method works well in practice and does not add any additional parameters. We will refer to this attention-level variant of our method as Syntactic Attention (`+SA`; Fig. 3.1, right).

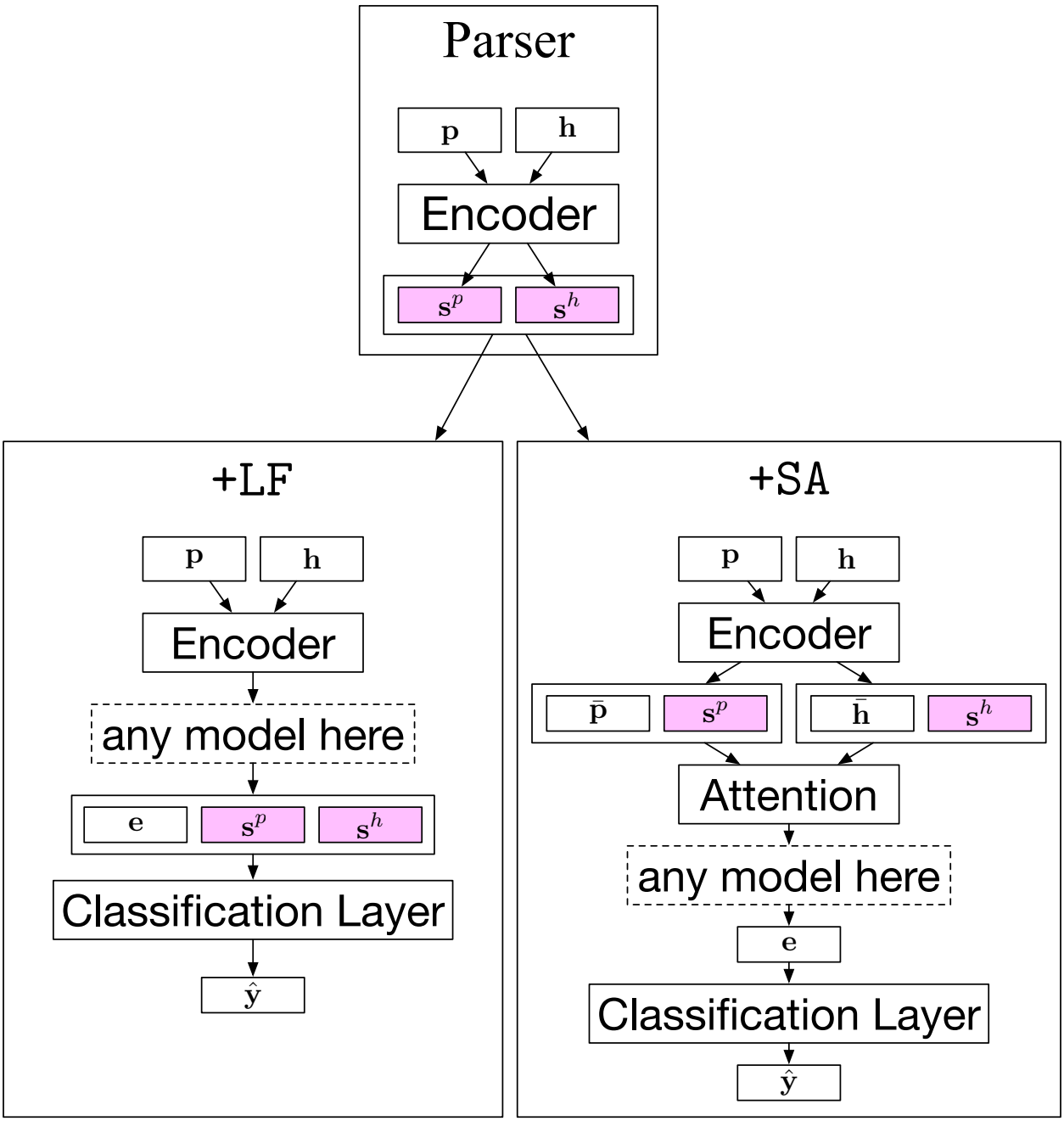

Figure 3.1: Our methods for incorporating syntactic representations, `+LF` (left) and `+SA` (right). $\boldsymbol{p}$ and $\boldsymbol{h}$ are the premise and hypothesis.

Chapter 4

EXPERIMENTS

We experiment on three NLI datasets: SNLI (Bowman et al., 2015), MNLI (Williams et al., 2018), and SciTail (Khot et al., 2018). We extend the decomposable attention and ESIM models with both `+LF` and `+SA` and extend BERT and MT-DNN with `+LF`.[1]

4.1 Datasets

We briefly summarize the datasets below.

4.1.1 SNLI

This dataset consists of 570k human-created and annotated sentence pairs; the premise sentences are drawn from the Flickr 30k corpus. A *contradiction* example from the dataset is shown below:

(1) ***p***: A man inspects the uniform of a figure in some East Asian country.
h: The man is sleeping.

4.1.2 MNLI

This dataset (433k pairs) was created using a similar methodology to SNLI, but with text drawn from ten domains. The development and test sets are separated into two categories: matched (MNLI-m; in-domain) and mismatched (MNLI-mm; cross-domain). A *neutral* example from the dataset is shown below:

[1]Multi-head self-attention (in BERT and MT-DNN) opens up many possibilities for `+SA` variants to be explored in the future.

(2) ***p***: I am a lacto-vegetarian.
h: I enjoy eating cheese too much to abstain from dairy.

4.1.3 SciTail

This dataset (27k pairs) is derived from science multiple-choice questions. Unlike SNLI and MNLI, the text was not deliberately generated for the corpus and there is no contradiction label. An *entailment* example from the dataset is shown below:

(3) ***p***: Beats are the periodic and repeating fluctuations heard in the intensity of a sound when two sound waves of very similar frequencies interfere with one another.
h: When waves of two different frequencies interfere, beating occurs.

4.2 Implementation Details

In all of our experiments, we use the deep biaffine attention model for dependency parsing by Dozat and Manning (2017);[2] in particular, we use the pretrained version from AllenNLP (Gardner et al., 2017) trained on the English Penn Treebank (Marcus et al., 1993) with Universal Dependencies (Nivre et al., 2016). The parser's parameters are frozen in all of our experiments.

For DA and ESIM, we use the implementations in AllenNLP; for BERT, we use the pretrained uncased $\text{BERT}_{\text{BASE}}$ from Hugging Face's PyTorch implementation;[3] and for MT-DNN we use the pretrained uncased $\text{BERT}_{\text{BASE}}$[4] MT-DNN model from the original authors.[5]

We perform random hyperparameter search for DA and ESIM and use hyperparameters reported in the original papers for BERT and MT-DNN. More details about our hyperpa-

[2]In preliminary experiments, we also explored the use of internal representations from a constituency parser; these experiments suggested that dependency parser representations worked better for our purposes.

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

[4]Note that state-of-the-art MT-DNN results on GLUE (and as a result, MNLI) use $\text{BERT}_{\text{LARGE}}$ which we do not experiment with.

[5]https://github.com/namisan/mt-dnn

	SNLI		SciTail		MNLI-m		MNLI-mm	
Model	dev.	test	dev.	test	dev.	test	dev.	test
DA	81.7	82.1	78.3	75.1	68.6	68.6	69.4	68.9
DA+LF	**84.9**	**84.8**	79.3	78.0	**71.6**	**71.6**	**72.4**	**71.0**
DA+SA	83.1	83.2	**82.8**	**78.2**	69.8	69.4	71.0	69.6
ESIM	**88.9**	87.9	80.7	76.2	**77.6**	**77.6**	77.4	**76.2**
ESIM+LF	**88.9**	87.7	83.7	78.0	77.2	77.3	**77.7**	76.1
ESIM+SA	88.4	**88.1**	**84.9**	**81.3**	76.9	77.1	77.5	75.8
BERT	90.5	89.9	**94.2**	90.8	84.3	84.7	**84.7**	**83.5**
BERT+LF	**90.6**	**90.5**	93.9	**92.8**	**84.7**	**84.9**	**84.7**	83.3
MT-DNN	**91.4**	**91.1**	**95.7**	94.0	84.2	84.1	**84.7**	83.3
MT-DNN+LF	91.3	**91.1**	95.1	**94.3**	**84.3**	**84.5**	84.6	**83.8**

Table 4.1: NLI dev./test accuracies. **Bold** is the top value on one dataset/architecture pair. Blue numbers represent accuracy increases and red numbers decreases, both relative to the syntax-free baseline of the same architecture.

rameters and tuning procedure can be found in Appendix A.

4.3 Results

Our experimental results are in Table 4.1. We find that adding representations from the syntax parser (+LF, +SA) always improves test accuracy on SNLI and SciTail across all baseline models. On MNLI, the results are mixed—DA, BERT, and MT-DNN improve on the matched subset, and only DA and MT-DNN improve on the mismatched subset.

SWRs always improve DA regardless of the method (+LR, +SA) or dataset. Despite the massive pretraining of BERT and MT-DNN, SWRs still improve BERT test accuracy on 3 of

4 test sets and MT-DNN test accuracy on 2 of 4 test sets. Our MT-DNN+LF model achieves new state-of-the-art results on SciTail.

Chapter 5

ANALYSIS

In this chapter, we perform an ablation study and probe our models with a diagnostic evaluation dataset.

5.1 Do Our Models Actually Use SWRs?

Dataset	+LF	+LF$_{\mathcal{N}}$	+SA	+SA$_{\mathcal{N}}$
SNLI	84.8	84.2 (-0.6)	83.2	75.6 (-7.6)
SciTail	78.0	75.2 (-2.8)	78.2	75.1 (-3.1)
MNLI-m	71.6	71.2 (-0.4)	69.4	63.0 (-6.4)
MNLI-mm	71.0	71.0 (-0.0)	69.6	62.9 (-6.7)

Table 5.1: DA ablation study test accuracies.

Dataset	DA	DA+LF$_{\mathcal{N}}$	DA+SA$_{\mathcal{N}}$
SNLI	82.1	84.2 (+2.1)	75.6 (-6.5)
SciTail	75.1	75.2 (+0.1)	75.1 (-0.0)
MNLI-m	68.6	71.2 (+2.6)	63.0 (-5.6)
MNLI-mm	68.9	71.0 (+2.1)	62.9 (-6.0)

Table 5.2: Test accuracies with random noise SWRs.

	SNLI		SciTail		MNLI-m		MNLI-mm	
Model	dev	test	dev	test	dev	test	dev	test
DA	81.7	82.1	78.3	75.1	68.6	68.6	69.4	68.9
DA+LF	84.9	84.8	79.3	78.0	71.6	71.6	72.4	71.0
DA+LF$_{\mathcal{N}}$	84.8	84.2	79.1	75.2	71.5	71.2	72.3	71.0
DA+SA	83.1	83.2	82.8	78.2	69.8	69.4	71.0	69.6
DA+SA$_{\mathcal{N}}$	75.6	75.6	69.9	75.1	63.7	63.0	63.7	62.9

Table 5.3: DA ablation study dev and test accuracies.

The DA models most consistently benefit from SWRs; one question is if they are simply taking advantage of the increased number of parameters. To examine this, we ablate the DA+LF and DA+SA models with random noise in place of the SWRs (denoted +LF$_{\mathcal{N}}$ and +SA$_{\mathcal{N}}$). In these experiments, $\mathbf{s} \sim \mathcal{N}(0, 1)$.

First, we observe that the test accuracies of DA+LF$_{\mathcal{N}}$ and DA+SA$_{\mathcal{N}}$ are less than or equal to their pretrained counterparts (DA+LF, DA+SA) on all datasets (Table 5.1). DA+SA is harmed most by random SWRs since +SA models directly use untransformed SWRs when calculating attention; that is, +SA models cannot learn to ignore the random noise.

We also observe that DA+LF$_{\mathcal{N}}$ performs *better* than DA on every dataset, while DA+SA$_{\mathcal{N}}$ performs *worse* than DA on every dataset except SciTail, where the test accuracy is unaffected (Table 5.2). This is consistent with our previous observation that +SA models cannot learn to ignore random noise while +LF models can. While it is surprising that DA+LF$_{\mathcal{N}}$ consistently improves performance over DA, we suspect this gain is the result of additional model parameters.

We report our full ablation study results in Table 5.3.

Heuristic	Definition	Example
Lexical overlap	Assume that a premise entails all hypotheses constructed from words in the premise	**The doctor** was **paid** by **the actor**. $\xrightarrow[\text{WRONG}]{}$ The doctor paid the actor.
Subsequence	Assume that a premise entails all of its contiguous subsequences.	The doctor near **the actor danced**. $\xrightarrow[\text{WRONG}]{}$ The actor danced.
Constituent	Assume that a premise entails all complete subtrees in its parse tree.	If **the artist slept**, the actor ran. $\xrightarrow[\text{WRONG}]{}$ The artist slept.

Table 5.4: This table by McCoy et al. (2019) shows the heuristics targeted by the HANS dataset, along with examples of incorrect entailment predictions that these heuristics would lead to.

5.2 Why Do SWRs Improve NLI?

The Heuristic Analysis for NLI Systems dataset (HANS; McCoy et al., 2019) is a controlled evaluation set for probing whether an NLI model learns fallible syntactic heuristics. HANS (30k pairs, labeled either *entailment* and *non-entailment*) is generated with templates that ensure the premise and hypothesis exhibit special syntactic relationships. The dataset is also balanced (i.e., random guessing should score around 50%). Table 5.4 describes the heuristics targeted by HANS and Table 5.5 presents examples from the dataset.

McCoy et al. (2019) show that NLI models learn heuristics around lexical overlap, subsequence relationships, and syntactic constituency between the premise and hypothesis;[1] they report that four NLI models trained on MNLI (including DA, ESIM, and BERT) score ∼50%

[1]These syntactic properties are commonly found in NLI datasets.

Heuristic	Premise	Hypothesis	Label
Lexical	The banker near the judge saw the actor.	The banker saw the actor.	E
overlap	The lawyer was advised by the actor.	The actor advised the lawyer.	E
heuristic	The doctors visited the lawyer.	The lawyer visited the doctors.	N
	The judge by the actor stopped the banker.	The banker stopped the actor.	N
Subsequence	The artist and the student called the judge.	The student called the judge.	E
heuristic	Angry tourists helped the lawyer.	Tourists helped the lawyer.	E
	The judges heard the actors resigned.	The judges heard the actors.	N
	The senator near the lawyer danced.	The lawyer danced.	N
Constituent	Before the actor slept, the senator ran.	The actor slept.	E
heuristic	The lawyer knew that the judges shouted.	The judges shouted.	E
	If the actor slept, the judge saw the artist.	The actor slept.	N
	The lawyers resigned, or the artist slept.	The artist slept.	N

Table 5.5: This table by McCoy et al. (2019) shows examples of sentences used to test the three heuristics in HANS. The *label* column shows the correct label for the sentence pair; *E* stands for *entailment* and *N* stands for *non-entailment*.

		Correct: *Entailment*			Correct: *Non-entailment*			
Model	Train Data	Lexical	Subseq.	Const.	Lexical	Subseq.	Const.	Avg.
DA	SciTail	93.4	89.1	90.3	7.1	9.9	11.0	50.1
DA+SA	SciTail	88.3	91.3	81.1	20.1	19.2	20.5	**53.4**
ESIM	SciTail	9.6	11.0	14.6	93.8	91.1	87.5	51.3
ESIM+SA	SciTail	78.0	84.1	84.4	29.2	19.2	16.8	**52.0**
BERT	SciTail	100	100	99.1	0.1	0.6	1.3	50.2
BERT+LF	SciTail	98.7	99.4	94.2	7.0	7.0	13.6	**53.2**
MT-DNN	MNLI-mm	99.0	100	99.8	53.1	2.8	3.8	59.7
MT-DNN+LF	MNLI-mm	99.0	99.8	99.8	58.3	3.8	5.2	**61.0**

Table 5.6: HANS accuracies of the model/dataset combintations whose NLI test accuracies improve the most with SWRs. The accuracies are split by the gold label and syntactic relationship between the premise and hypothesis.

overall and close to 0% on the *non-entailment* label. Examples of sentence pairs exhibiting

We evaluate a subset of our models on HANS. In particular, we choose the model/dataset combinations whose test accuracies improve the most with SWRs: DA+SA on SciTail (+3.1%), ESIM+SA on SciTail (+5.1%), BERT+LF on SciTail (+2.0%), and MT-DNN+LF on MNLI-mm (+0.5%). We find SWRs improve overall accuracy on HANS (1–3%) over their non-SWR counterparts for all four models listed above. We also observe that SWRs significantly reduce the models' reliance on fallible syntactic heuristics—that is, the accuracies between the *entailment* and *non-entailment* labels are far more balanced. Our full experimental results on HANS can be found in Table 5.6.

Chapter 6

RELATED WORK

There is a long line of work on incorporating syntactic features for semantic tasks, including NLI. Some earlier approaches include learning syntactic rules indicative of entailment (e.g., Bar-Haim et al., 2007; Mehdad et al., 2010) or features derived from tree transformations between premise and hypothesis (Heilman and Smith, 2010; Wang and Manning, 2010). More recent research has incorporated syntactic structures into neural architectures (e.g., tree LSTMs in Chen et al. 2017; graph-based architecture in Khot et al. 2018).

Other recent work has experimented with incorporating syntactic learning objectives. Strubell et al. (2018) and Swayamdipta et al. (2018) used multitask learning of syntactic and semantic tasks to transfer syntactic features, improving performance on semantic role labeling and, in the latter case, coreference resolution.

Liu et al. (2019a) and Hewitt and Manning (2019) showed that, to some degree, contextual word representations encode linguistic and syntactic knowledge. Our experimental results with BERT and MT-DNN motivate future work to understand exactly what knowledge contextual word representations contain and the effects of fine-tuning models like BERT.

Chapter 7

CONCLUSION

In this work, we introduced a simple and broadly applicable method for incorporating syntax into NLI models, through the use of contextual vector representations from a pre-trained parser. We demonstrated that our method often improves accuracy for four NLI models (DA, ESIM, BERT, MT-DNN) across three standard NLI datasets. Our findings demonstrate the usefulness of syntactic information in semantic models and motivate future research into syntactically informed models.

BIBLIOGRAPHY

Roy Bar-Haim, Ido Dagan, Iddo Greental, and Eyal Shnarch. 2007. Semantic inference at the lexical-syntactic level. In *AAAI*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *EMNLP*.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for natural language inference. In *ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Timothy Dozat and Christopher D Manning. 2017. Deep biaffine attention for neural dependency parsing. In *ICLR*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. AllenNLP: A deep semantic natural language processing platform. arXiv:1803.07640.

Michael Heilman and Noah A. Smith. 2010. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *NAACL-HLT*.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *NAACL-HLT*. Association for Computational Linguistics.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. SciTail: A textual entailment dataset from science question answering. In *AAAI*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv:1412.6980.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *TACL*, 4:313–327.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual representations. In *NAACL-HLT*.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019b. Multi-task deep neural networks for natural language understanding. arXiv:1901.11504.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19:313–330.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *NeurIPS*.

R Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *ACL*.

Yashar Mehdad, Alessandro Moschitti, and Fabio Massimo Zanzotto. 2010. Syntactic/semantic structures for textual entailment recognition. In *NAACL-HLT*.

Richard Montague. 1970. Universal grammar. *Theoria*, 36(3):373–398.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *LREC*.

Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *EMNLP*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *EMNLP*.

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

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Mark Steedman and Jason Baldridge. 2011. Combinatory categorial grammar. *Non-Transformational Syntax: Formal and explicit models of grammar*, pages 181–224.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *EMNLP*.

Swabha Swayamdipta, Sam Thomson, Kenton Lee, Luke Zettlemoyer, Chris Dyer, and Noah A. Smith. 2018. Syntactic scaffolds for semantic structures. In *EMNLP*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

Alex Wang, Amapreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR*.

Mengqiu Wang and Christopher Manning. 2010. Probabilistic tree-edit models with structured latent variables for textual entailment and question answering. In *COLING*.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL-HLT*.

Appendix A

HYPERPARAMETERS AND TUNING PROCEDURE

In this section, we report our hyperparameters and describe our tuning procedure.

A.1 Word Embeddings

The parser uses 100-dimensional GloVe word embeddings (Pennington et al., 2014) trained on Wikipedia/Gigaword while the rest of our models use 300-dimensional GloVe word embeddings.

A.2 Optimizer

We train DA, ESIM, and BERT with Adam (Kingma and Ba, 2014), and for MT-DNN, we use the Adamax variant of Adam following Liu et al. (2019b).

A.3 Decomposable Attention Model

For the DA baseline and each variant, we randomly sample 30 configurations and choose the best performing dev. configuration on SciTail. We sometimes round the hyperparameters for ease of implementation. We use the hyperparameters chosen on SciTail for all other datasets. Each hyperparameter is sampled from a range, either uniformly or log-uniformly (Tab. A.1). Our chosen DA hyperparameters are reported in Table A.2.

A.4 ESIM

For the ESIM baseline and each variant, we randomly sample 10 configurations (Tab. A.3, A.4).

Param.	Range	Sampling Method
Learning Rate	$[1 \times 10^{-6}, 0]$	Log-Uniform
Att. FF HD	[100, 300]	Uniform
Comp. FF HD	[100, 400]	Uniform
Agg. FF HD	[100, 400]	Uniform
Att. FF Dropout	[0.2, 0.7]	Uniform
Comp. FF Dropout	[0.2, 0.7]	Uniform
Agg. FF Dropout	[0.2, 0.7]	Uniform

Table A.1: DA hyperparameter sampling ranges and methods. FF stands for feed-forward and HD stands for hidden dimensions.

Param.	DA	DA+LF	DA+SA
Learning Rate	3×10^{-4}	3×10^{-4}	3×10^{-4}
Att. FF HD	295	250	295
Comp. FF HD	108	400	108
Agg. FF HD	172	150	295
Att. FF Dropout	0.29	0.3	0.29
Comp. FF Dropout	0.34	0.3	0.34
Agg. FF Dropout	0.54	0.3	0.54

Table A.2: Chosen DA hyperparameters. FF stands for feed-forward and HD stands for hidden dimensions.

Param.	Range	Sampling Method
Learning Rate	$[1 \times 10^{-4}, -1]$	Log-Uniform
Model Dropout	[0.2, 0.7]	Uniform
Output FF Dropout	[0.2, 0.7]	Uniform

Table A.3: ESIM hyperparameter sampling ranges and methods. FF stands for feed-forward.

Param.	DA	DA+LF	DA+SA
Learning Rate	3×10^{-4}	3×10^{-4}	3×10^{-4}
Att. FF HD	295	250	295
Comp. FF HD	108	400	108
Agg. FF HD	172	150	295
Att. FF Dropout	0.29	0.3	0.29
Comp. FF Dropout	0.34	0.3	0.34
Agg. FF Dropout	0.54	0.3	0.54

Table A.4: Chosen ESIM hyperparameters. FF stands for feed-forward.

A.5 BERT and MT-DNN

We use the hyperparameters reported by Devlin et al. (2019) and Liu et al. (2019b). The only exception is that for MT-DNN+LF, we find that in some cases a longer warmup (0.3) improves performance.