Is Semantic-aware BERT more Linguistically Aware?  
A Case Study on Natural Language Inference

Ling Liu*  
University of Colorado  
ling.liu@colorado.edu

Ishan Jindal  
IBM Research  
ishan.jindal@ibm.com

Yunyao Li†  
Apple  
yunyaoli@apple.com

Abstract

Recent work has shown that predicate-argument label representations from semantic role labeling (SRL) can be concatenated with BERT representations to improve natural language understanding tasks such as natural language inference (NLI) and reading comprehension. Two natural questions that arise are whether infusing SRL representations with BERT 1) improves model performance and 2) increases the model’s linguistic awareness. This paper aims at answering both questions with a case study on the NLI task. We start by analyzing whether and how infusing SRL information helps BERT learn linguistic knowledge. We compare model performance on two benchmark datasets, SNLI and MNLI. We also conduct in-depth analysis on two probing datasets, Breaking NLI and HANS, which contain abundant examples where SRL information is expected to be helpful. We found that combining SRL representations with BERT representations does outperform BERT-only representations in general, with better awareness in lexical meaning and world knowledge but not in logic knowledge. We also found that infusing SRL information via predicate-wise concatenation with BERT word representations followed by an interaction layer is more effective than sentence-wise concatenation.†

1 Introduction

Recent advances in Transformer-based language models like BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019b), have surpassed the performance of non-expert humans when fine-tuned and evaluated on various benchmark datasets such as GLUE (Wang et al., 2018), SuperGLUE (Wang et al., 2019a). However, performing well on benchmark datasets does not mean the model has achieved human-like natural language understanding (NLU) competence (Naik et al., 2018; Talman and Chatzikyriakidis, 2019). It has been found that the transformer models when fine-tuned for a particular tasks are prone to adopt shallow heuristics (McCoy et al., 2019). For instance, in a natural language inference (NLI) system — a task of determining whether a premise sentence entails a hypothesis sentence (Dagan et al., 2005; MacCartney and Manning, 2009), a model is more likely to assign a label “contradiction” if the word not is present in the hypothesis and assign a label “entailment” if all words in a hypotheses are present in the premise (Naik et al., 2018).

Consider the following sentence pair where hypothesis contradicts premise:

P: The lawyers were recommended by the doctor
H: The lawyers recommended the doctor

A RoBERTa model fine-tuned on MNLI dataset, benchmark NLI dataset (Williams et al., 2018), completely ignores the semantic meaning of these sentences and predicts entailment\(^2\). This wrong prediction indeed happened because the model learns to generalize over the simple heuristic present in the training datasets that is predicting entailment when there is 100% lexical overlap between premise and hypothesis. At this point, an important question arises: Does providing semantic information to the language models in the form of external knowledge enhance models’ linguistic knowledge, if yes, how?

Semantic role labeling (SRL) is a shallow semantic parsing task that identifies “who did what to whom when, where etc” for each predicate in a sentence, called predicate-argument structure. SRL represents the semantic meaning of a sentence and

\(^2\)https://demo.allennlp.org/textual-entailment/roberta-mnli/s/the-lawyers-were-recommended-doctor/P8E4F0W1U9
this information is essential to natural language understanding (Palmer et al., 2005; He et al., 2017; Kasai et al., 2019; Jindal et al., 2020; Marcheggiani and Titov, 2020). For the sentence pair above, The semantic roles associated with the verb *recommend* are as follows:

**P:** [ARG1 : The lawyers] were [v : recommended] by [ARG0 : the doctor]

**H:** [ARG0 : The lawyers] [v : recommended] [ARG1 : the doctor]

This semantic information clearly distinguishes the *Recommender ARG0* in the premise and the hypothesis: the *doctor* is the *Recommender* in the premise and *lawyer* is the *Recommender* in the hypothesis. In this example, both the premise and the hypothesis have identical predicate-argument structures with the same predicate but swapped arguments. We expect that infusing SRL information would be helpful in such cases as it highlights the difference in the arguments, and thus the model is directed to the lexical meaning difference encoded by the tokens to correctly classify the example.

Recently, Zhang et al. (2019b) propose SemBERT to incorporate representations of semantic role labels (SRLs) into the BERT representations. In SemBERT, both representations were processed independently and get concatenated for fine-tuning. This concatenation alone shows slight improvement over BERT on NLU benchmark datasets but fails to improve the model’s linguistic awareness as shown by its poor performance on probing datasets. In this work, we propose a simple yet more effective improvement over SemBERT to improve the model’s awareness of various language phenomena such as lexical knowledge, world knowledge. Specifically, our main contributions include:

- We present detailed analysis of SRL-infused models on two probing datasets, Breaking NLI (Glockner et al., 2018) and HANS (McCoy et al., 2019) to understand what cases benefit or suffer from the infused SRL information.
- We propose Linguistic BERT (LingBERT) that concatenates predicate-wise SRL representations with BERT representations followed by an interaction layer. We demonstrate its effectiveness over existing baselines on two commonly used NLI benchmark datasets, SNLI and MNLI.
- We found that LingBERT does learn lexical and world knowledge better. However, how to leverage SRL information to improve the model’s awareness of other linguistic phenomena remains an open challenge.

## 2 Motivation

Semantic role labeling in the PropBank format labels syntactic constituents for each predicate in the sentence with the corresponding generalized prototypical roles such as ARG0, ARG1 etc. Therefore, though shallow, SRL provides explicit syntactic and semantic knowledge.

Existing studies (Naik et al., 2018; Talman and Chatzikyriakidis, 2019) point out that though large-size pre-trained language models can achieve good performance on classification tasks, such models are still far from a real understanding of language. Does incorporating SRL information explicitly at the fine-tuning stage increase the model’s linguistic awareness and lead to performance improvement?

This paper aims at answering this question by comparing the performance of BERT with models incorporating semantic role labels into BERT on two benchmark NLI datasets, SNLI and MNLI. We further analyze whether incorporating the additional SRL information increases the models’ linguistic awareness by analyzing model performance on two probing datasets, Breaking NLI and HANS, which contain abundant cases where SRL information is expected to be helpful.

### 2.1 Issues with NLI Benchmark Datasets

Two commonly-used NLI benchmark datasets, SNLI\(^3\) (Bowman et al., 2015) and MNLI\(^4\) (Williams et al., 2018) are created via crowdsourcing. The models fine-tuned on these datasets perform well on their corresponding test sets. However, such models tend to adopt simple heuristics from the training data that are effective only for most frequent example types (McCoy et al., 2019). In other words, a model fine-tuned on SNLI training set may perform excellent (91.0% accuracy) on the corresponding test set but perform poorly (46.3% accuracy, in Table 5) when tested on a specific heuristic. Therefore, *it is imperative to test whether the fine-tuned models generalize over certain heuristics.*

---

\(^3\)https://nlp.stanford.edu/projects/snli/

\(^4\)https://cims.nyu.edu/~sbowman/multinli/
The family is drinking wine

\[ \text{P} \quad \text{The family is drinking wine} \]

\[ \text{B-ARG0 I-ARG0 O B-V B-ARG1} \]

The family is drinking vodka

\[ \text{H}_1 \quad \text{The family is drinking vodka} \]

\[ \text{B-ARG0 I-ARG0 O B-V B-ARG1} \]

The family is drinking gin

\[ \text{H}_2 \quad \text{The family is drinking gin} \]

\[ \text{B-ARG0 I-ARG0 O B-V B-ARG1} \]

Table 1: World knowledge heuristic examples from Breaking NLI dataset. SRLs (in blue) are expected to be helpful for such examples. The ground-truth NLI label for the premise and each hypothesis is contradiction. However, BERT predicts neutral for P-H$_1$, and contradiction P-H$_2$. Our proposed method classifies both cases correctly.

### 2.2 Probing Information

Various probing datasets have been developed (Glockner et al., 2018; McCoy et al., 2019) to analyze whether the trained model has learned the linguistic knowledge and reasoning generalization that humans resort to for the same task. Though the model has learned some linguistic features (Goldberg, 2019; Liu et al., 2019a; Hewitt and Manning, 2019; Clark et al., 2019; Lin et al., 2019; Tenney et al., 2019; Warstadt and Bowman, 2020; Manning et al., 2020; Ettinger, 2020; Michael et al., 2020), the poor performance of fine-tuned BERT models on the probing datasets indicates that it is not enough linguistic knowledge. We evaluate the fine-tuned models on two probing datasets: Breaking NLI$^5$ (Glockner et al., 2018) and HANS$^6$ (McCoy et al., 2019). Though not directly created to probe about SRL information, the two datasets contain abundant examples on different heuristics for which SRL information is expected to be helpful.

#### 2.2.1 Lexical Meaning and World Knowledge

Breaking NLI provides examples to test whether the trained model has learned proper lexical meaning and world knowledge. It contains instances where the premise and hypothesis are identical except for one token as shown in Table 1. Infusing SRL information is expected to be helpful for this dataset because the extra predicate-argument information enforces the BERT model to focus more on distinctions encoded in the lexical meaning.

#### 2.2.2 Lexical Overlap Heuristic

Lexical overlap is the first type of heuristic which the HANS dataset intends to probe about. It contains examples where hypotheses are constructed from all the words in premises. Both Breaking NLI and HANS provide examples to test this heuristic. Table 2 lists examples from the lexical overlap heuristics in Row 1. As can be seen, both the premise and the hypothesis have identical predicate-argument structures, where both have the same predicate with only the arguments swapped. We would expect infusing SRLs to be helpful in this case because it highlights the similarity in the predicate-argument structure of the text and thus the model is directed to the lexical meaning difference encoded by the tokens in order to correctly classify the example.

#### 2.2.3 Subsequence Heuristic

Subsequence heuristic is the second type of heuristic the HANS dataset intends to probe about. In the subsequence heuristic example (Table 2 Row 2), the predicate-argument structures of the premise and the hypothesis are quite different. Here the premise has two predicates whereas the hypothesis has only one. This kind of tricky cases where the hypothesis is a sub-string of the premise is relatively hard for NLI models to learn. We also expect the predicate-argument structure to be helpful here because it highlights the syntactic difference of identical tokens: the actors is ARG1 for the predicate believe in the hypothesis while it is only a component of ARG1 for the same predicate in the premise.

#### 2.2.4 Constituent Heuristic

Constituent heuristic is the third type of heuristic the HANS dataset intends to probe about. It contains examples where the premise entails all complete subtrees of its parse tree as shown in Table 2 Row 3. In comparison to other heuristics, the generalization over constituent heuristic examples requires more logical information rather than just predicate-argument structures. For the PP_on_subject sub-case of constituent, span-base SRL actually introduces more misleading information. Therefore, we do not expect the SRL information to be helpful for this case.

### 3 How to Incorporate Semantic Knowledge?

In last section we describe various heuristics where vanilla transformer model does not generalize well.

---

5https://github.com/BIU-NLP/Breaking_NLI

6https://github.com/tommccoy1/hans
when fine-tuned on NLI datasets. In this section, we describe the limitations of existing semantic aware transformer models and describe the strategy to further improve their performance.

We follow Devlin et al. (2019) to formulate textual entailment as a classification problem and design the input to the BERT model as \([\text{CLS}] P_i [\text{SEP}] H_i [\text{SEP}]\), where \(P_i = \{w^1_{p1}, w^2_{p2}, \ldots, w^n_{pK}\}\) is the premise and \(H_i = \{w^1_{h1}, w^2_{h1}, \ldots, w^n_{hK}\}\) is the hypothesis. The final hidden layer vector representation of the first symbol in the sequence [CLS] is used as input to the classification layer at the fine-tuning stage.

Zhang et al. (2019b) propose SemBERT to concatenate SRL embeddings with BERT representations. In addition to vanilla BERT embeddings, they apply an out-of-box span-based semantic role labeling model to label the premise and hypothesis text with SRLs in PropBank format. If the premise or the hypothesis is associated with multiple predicate-argument structures, a linear layer is used to convert the multiple semantic role labels associated with each token into one SRL embedding representation. Then they concatenate the BERT representation with the SRL representation tokenwise, and use the first token \([\text{CLS}]\) representation for classification. We refer to the SemBERT approach as semantic role labeling model as SemBERT to label the premise and the hypothesis.\(^7\) We obtain multiple predicate-argument structures for both the premise and the hypothesis, equal to the number of predicates in respective text. For a sentence \(S = \{w^1, w^2, \ldots, w^n\}\) containing \(K\) predicates, we obtain \(K\) predicate-argument structures \([(l^1_1, l^2_1, \ldots, l^n_1), \ldots, (l^1_K, l^2_K, \ldots, l^n_K)\)] \(l^j_k\) represents a semantic role label for token \(j\) specific to the \(k\)th predicate. We represent the SRLs as embeddings and use a lookup table to map the SRLs to vectors, which are learned during training. Therefore, for sentence \(S\) we obtain \(\{v^1_1, v^2_1, \ldots, v^n_1\}, \ldots, \{v^1_K, v^2_K, \ldots, v^n_K\}\) as SRL representations.

For each token \(w^j\) in sentence \(S\) we obtain its fused embedding based on its BERT representation and the corresponding SRL representations as \(\{e^j \circ v^1_1, e^j \circ v^2_1, \ldots, e^j \circ v^n_1\}\), where \(\circ\) represents concatenation. For completeness and simplicity, we assume a “○” label for [CLS] and [SEP] tokens. This fused embedding is passed through a linear layer to increase interaction among

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>non-entailment</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lexical overlap</td>
<td>P: The judge encouraged the athlete</td>
<td>B-ARG1 I-ARG0 B-V B-ARG1 I-ARG0 0</td>
</tr>
<tr>
<td></td>
<td>H: The athlete encouraged the judge</td>
<td>B-ARG0 I-ARG0 B-V B-ARG1 I-ARG0 0</td>
</tr>
<tr>
<td>Subsequence</td>
<td>P: The scientists believed the actors saw the artists</td>
<td>B-ARG1 I-ARG0 B-V B-ARG1 I-ARG0 0</td>
</tr>
<tr>
<td></td>
<td>H: The scientists believed the artists</td>
<td>B-ARG0 I-ARG0 B-V B-ARG1 I-ARG0 0</td>
</tr>
<tr>
<td>Constituent</td>
<td>P: Hopefully the president introduced the doctors</td>
<td>B-ARG1 I-ARG0 B-V B-ARG1 I-ARG0 0</td>
</tr>
<tr>
<td></td>
<td>H: The president introduced the doctors</td>
<td>B-ARG0 I-ARG0 B-V B-ARG1 I-ARG0 0</td>
</tr>
</tbody>
</table>

Table 2: non-entailment examples from HANS with SRLs (in blue) tags.
BERT and SRL representations, and the representation of the first input token [CLS] is then used as an input to the classifier.

Our model differs from SemBERT (Zhang et al., 2019b) in two aspects (see Figure 1 for an illustration of our model). First, LingBERT concatenates the BERT representation for each token with the SRL representation for each predicate-argument structure respectively, rather than fuses SRLs with each token disregarding the lexical meaning of the predicate as SemBERT does. LingBERT is expected to be more effective than SemBERT because each predicate-argument structure is labeled with respect to the predicate (Palmer et al., 2005). Second, in SemBERT, the interaction between the BERT representation and the SRL embedding is only by concatenation. In contrast, LingBERT adds a linear layer after concatenating BERT representations and SRL representations to allow the interaction between them.

4 Experiments and Results

4.1 Model Setup

The pre-trained English BERT-base-uncased model is used in all experiments. For BERT fine-tuning, we used the Transformer (Vaswani et al., 2017) implementation by Wolf et al. (2019)\(^8\). For SemBERT, the implementation by Zhang et al. (2019b)\(^9\) is used. LingBERT is implemented based on SemBERT. We adopt the same hyperparameters as SemBERT in all experiments. We trained each model multiple times and report the average accuracy. Following Zhang et al. (2019b), we obtain span-based\(^10\) SRLs from AllenNLP (Gardner et al., 2018).

4.2 Overall Performance

As shown in Tables 3 and 4, both SRL-infused models, SemBERT and LingBERT, outperform BERT on SNLI test set and MNLI development sets. This result confirms that infusing SRL representations with BERT representations does improve the model’s performance on NLI.

4.2.1 SNLI

LingBERT outperforms all semantic aware models including SemBERT on SNLI test set and on both the probing datasets, Breaking NLI and HANS (see Table 3). The evident advantage of LingBERT over SemBERT confirms that predicate-wise concatenation with interaction is more effective than sentence-wise concatenation.

4.2.2 MNLI

Similar to SNLI we observe that the semantic knowledge helps improve the model performance over simple BERT baselines when fine-tuned on MNLI dataset, in Table 4. We also observe a slight performance drop for LingBERT when compared with SemBERT on MNLI dev set, we explain this behaviour in Section 5.2.

4.3 Performance on Probing Datasets

4.3.1 Breaking NLI

Independent of training dataset, LingBERT outperforms BERT and SemBERT, achieving SoTA

---

\(^8\)https://github.com/huggingface/transformers
\(^9\)https://github.com/cooelf/SemBERT
\(^10\)We obtain BIO tags for each token in the sentence. Throughout this work we use span-based SRLs.
accuracy 94.04% on Breaking NLI dataset. This indicates that LingBERT has learned lexical meaning and world knowledge better.

As mentioned in Section 2.2, the premise and hypothesis in this dataset have highly similar or identical predicate-argument structures. Table 1 provides examples where LingBERT consistently predicts correct labels while BERT performs inconsistently. Though in both examples only the objects of the verb differ, BERT model predicts neutral for the P-H₁ and predicts contradiction for the P-H₂. The key to the correct prediction is to know the lexical meaning difference between wine and vodka, and between wine and gin. LingBERT, with SRL infused, captures the similarity between the premise and hypothesis, and learns the lexical meaning and world knowledge distinction to make the correct prediction.

4.3.2 HANS

This dataset was created by manipulating the syntactic structure and constituents with prescribed rules. Better performance of a model on HANS means that the trained model is less vulnerable to lexical overlap, subsequence, or constituent heuristics present in the training data. As analyzed Section 2.2, we expect SRL information to help with cases of the lexical overlap and subsequence heuristics.

From Tables 3 and 4, we made the following observations for HANS dataset:

1. While LingBERT consistently outperforms SemBERT on HANS, it only outperforms BERT when fine-tuned on SNLI.
2. The BERT model fine-tuned on MNLI outperforms the BERT model fine-tuned on SNLI on HANS. However, SemBERT and LingBERT show the opposite tendency.

The detailed analysis of these observations is presented in the subsequent sections.

5 Detailed Analysis

In Section 2, we observe that there are some heuristics where infusing SRL information is useful. In this section, we analyze the performance of the SRL infused models on different heuristics in detail and analyze the effect of different training datasets on model performance.

5.1 Heuristic Level Analysis

When fine-tuned on SNLI (see Table 3), LingBERT achieves the best overall performance on HANS, followed by BERT and SemBERT. Considering that every model classifies most examples as entailment and achieves an accuracy of ~99% for the ground-truth entailment examples, we examine the models trained on SNLI training set and evaluated on the HANS non-entailment examples for each heuristic type and its sub-cases to better interpret their performance. Details of model performance for these examples are summarized in Table 5.

5.1.1 Lexical Overlap Heuristic

As expected, LingBERT outperforms other models by a big margin on every sub-case of the lexi-
Table 5: Performance on HANS non-entailment examples by models fine-tuned on SNLI. Examples in black and normal font are where BERT made wrong predictions and LingBERT made correct predictions. Examples in blue and italics are where none of the three models made the correct prediction. The last three columns are the accuracy in % on the non-entailment examples by BERT, SemBERT, and LingBERT respectively.

<table>
<thead>
<tr>
<th>HANS Heuristics</th>
<th>non-entailment Examples</th>
<th>BERT</th>
<th>SemBERT</th>
<th>LingBERT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lexical Overlap Heuristic</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ln_conjunction</td>
<td>P1: The authors recognized the president and the judges . H1: The judges recognized the president .</td>
<td>46.53</td>
<td>43.02</td>
<td>54.40</td>
</tr>
<tr>
<td>ln_passive</td>
<td>P2: The lawyers were recommended by the doctor . H2: The lawyers recommended the doctor .</td>
<td>40.93</td>
<td>33.57</td>
<td>50.63</td>
</tr>
<tr>
<td>ln_preposition</td>
<td>P3: The senators behind the lawyer contacted the student . H3: The student contacted the senators .</td>
<td>17.90</td>
<td>30.77</td>
<td>33.00</td>
</tr>
<tr>
<td>ln_relative_clause</td>
<td>P4: The student who the senators thanked stopped the scientist . H4: The scientist stopped the student .</td>
<td>58.37</td>
<td>49.20</td>
<td>60.37</td>
</tr>
<tr>
<td>ln_subject/object_swap</td>
<td>P5: The student saw the managers . H5: The managers saw the student .</td>
<td>46.67</td>
<td>40.20</td>
<td>52.63</td>
</tr>
<tr>
<td>ln_relative_clause P</td>
<td>P6: The students who the senators thanked stopped the scientist . H6: The student contacted the students .</td>
<td>67.77</td>
<td>61.37</td>
<td>75.37</td>
</tr>
<tr>
<td>Subsequence Heuristic</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sn_NP/S</td>
<td>P7: The author heard the presidents recommended the secretary . H7: The author heard the presidents .</td>
<td>4.92</td>
<td>3.69</td>
<td>4.01</td>
</tr>
<tr>
<td>sn_NP/Z</td>
<td>P8: Although the managers hid the actors saw the athlete . H8: The managers hid the actors .</td>
<td>0.70</td>
<td>0.03</td>
<td>0.53</td>
</tr>
<tr>
<td>sn_PP_on_subject</td>
<td>P9: The student near the secretaries supported the judges . H9: The secretaries supported the judges .</td>
<td>9.67</td>
<td>6.43</td>
<td>5.27</td>
</tr>
<tr>
<td>sn_past_participle</td>
<td>P10: The artist avoided the author paid in the laboratory . H10: The author paid in the laboratory .</td>
<td>9.03</td>
<td>6.9</td>
<td>7.83</td>
</tr>
<tr>
<td>sn_relative_clause_on_subject</td>
<td>P11: The scientists who introduced the senator avoided the actor . H11: The senator avoided the actor .</td>
<td>0.80</td>
<td>0.27</td>
<td>0.80</td>
</tr>
<tr>
<td>Constituent Heuristic</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cn_adverb</td>
<td>P12: Hopefully the presidents introduced the doctors . H12: The presidents introduced the doctors .</td>
<td>5.2</td>
<td>2.44</td>
<td>3.03</td>
</tr>
<tr>
<td>cn_after_if_clause</td>
<td>P13: Unless the professor slept , the tourist saw the scientist . H13: The tourist saw the scientist .</td>
<td>0.20</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>cn_disjunction</td>
<td>P14: The professor slept , the tourist saw the scientist . H14: The tourist saw the president .</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>cn_embedded_under_if</td>
<td>P15: If the doctors mentioned the judge , the president thanked the student . H15: The doctors mentioned the judge .</td>
<td>0.33</td>
<td>0.03</td>
<td>0.00</td>
</tr>
<tr>
<td>cn_embedded_under_verb</td>
<td>P16: The lawyers believed that the tourists shouted . H16: The tourists shouted .</td>
<td>25.3</td>
<td>12.2</td>
<td>15.1</td>
</tr>
<tr>
<td></td>
<td>P17: The lawyers believed that the tourists shouted . H17: The tourists shouted .</td>
<td>0.13</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

cal overlap heuristic (see Table 5 Row 1), indicating the contribution of effectively infusing SRL representations. In contrast, SemBERT, which also incorporates SRL representations, fails to outperform BERT for all sub-cases except for one (In_preposition). This observation supports that predicate-wise concatenation with interaction is more effective to infuse SRL than sentence-wise concatenation.

Column 2 lists examples for each sub-case. The premise and hypothesis for all examples consist of one clause (except the premise for the ln_relative_clase example) with very similar predicate-argument structures. The similarity and simplicity of the predicate-argument structures of the premise and the hypothesis seem to be related to the contribution of SRL in LingBERT. When the premise and hypothesis have similar predicate-argument structures, incorporating SRL representations emphasizes the similarity between them, thus enforces BERT to learn to distinguish lexical meaning and world knowledge.

5.1.2 Subsequence Heuristic

All models perform poorly for the subsequence heuristic as shown in Table 5 Row 2. Overall, BERT outperforms SemBERT and LingBERT, though not a single model improves consistently over all the sub-cases. All examples (except sn_relative_clause_on_subject and sn_PP_on_subject) were created to confuse the model by leveraging the syntactic flexibility of the predicates. For instance, P1 - H1 leverages the syntactic flexibility of hear that the ARG0 of this verb can be a clause as in P1 or a noun phrase as in H1. P2-H2 leverages the syntactic flexibility of hide that this verb can be intransitive as in P2 or transitive
 Against our expectations discussed in the section on probing informa, SRL-infused models almost consistently perform worse than BERT. The poor performance of such models demands better methods of incorporating SRL embeddings with BERT representations.

Note that on sn_PP_on_subject sub-case, we observe the expected performance drop when SRL representations are infused with BERT representations. This drop is because the span-based predicate-argument structure introduces misleading information. For example, the predicate-argument structure of the premise of P3-H3 in Table 5 Row 2 is [ARG0 The student near the secretaries] [v supported] [ARG1 the judges] and that for the hypothesis is [ARG0 The secretaries] [v supported] [ARG1 the judges]. In both the premise and the hypothesis, the secretaries is within the ARG0 span, though in the premise it’s only the modifier of the ARG0 which is not distinguished in span-based semantic role labeling. This information misleads the span-based SRL infused models to predict entailment opposed to non-entailment. We suspect head-based SRL information may improve the performance on the examples of this sub-case, which can be explored in future work.

5.1.3 Constituent Heuristic

As discussed earlier, we do not expect to observe LingBERT generalize over this heuristic, because SRL can not provide the critical logical knowledge to improve model performance on this heuristic. In Table 5 Row 3, all models perform poorly on this heuristic with basically no difference. For all the constituent heuristic examples, the truth condition of the premise is changed either by adding certain particular words or by using certain syntactic structures with particular connecting words. The words through which the premise’s truth condition gets reversed are marked out in bold font in these examples. The poor performance of LingBERT and SemBERT for these examples indicates that incorporating semantic role labels cannot help BERT learn logic knowledge as expected.

5.2 Effect of Training Data

Tables 3 shows that there is performance boost on HANS in LingBERT when it is fine-tuned on SNLI. However, from the performance on HANS presented in Table 4, we observe the opposite for fine-tuning with MNLI. To understand why there is the opposite pattern, we examine how the following factors associated with training data may influence the effect of infusing SRL representations: (1) complexity of the training data, and (2) quality of semantic role labels.

5.2.1 Complexity of Training Dataset

From Tables 3 and 4, we observe that the performance of the BERT model on HANS varies largely with the training dataset. Therefore, we start by analyzing the effect of training data complexity on the fine-tuned BERT model. Since MNLI has premises of more diversified genres with longer and more complex sentences than SNLI, we hypothesize that the BERT model fine-tuned on longer and more complex sentences is less likely to overfit the three types of heuristics in HANS. Conversely, the BERT model fine-tuned on SNLI is more likely to overfit to these heuristics. Hence, the contribution of the SRL representations incorporated in the BERT representations may outweigh the noise it introduces when fine-tuned on SNLI in comparison to MNLI where BERT representations alone better generalize over such heuristics. To verify the hypothesis that longer and more complex NLI training data tend to have fewer supporting cases for HANS heuristics, we fine-tune BERT on the combined ANLI training set. ANLI (Nie et al., 2020) is a newly created NLI dataset with much longer and more complex premises from more diversified domains than MNLI. However, BERT model fine-tuned on this dataset achieves only 56.44% accuracy on HANS, even lower than BERT fine-tuned on SNLI (58.83%). Therefore, we rule out the possibility that the complexity of training data is the only contributor towards the better performance of MNLI fine-tuned BERT model.

5.2.2 Quality of SRL

Similar to SNLI fine-tuned LingBERT we expect to see performance improvement on MNLI fine-tuned LingBERT over BERT. However, though LingBERT outperform SemBERT, we observe an obvious performance drop when compared to BERT. To understand it further we hypothesize that the noise in the SRL process may get carried over to the downstream NLI task with LingBERT and led to the performance drop with the infusion of SRL representations. To test this hypothesis, we combined the labeled SNLI training data and the la-

11https://dl.fbaipublicfiles.com/anli/anli_v1.0.zip
beled MNLI training data respectively with the CoNLL 2012 SRL training data (Pradhan et al., 2013)\textsuperscript{12} to train an in-house SRL model. We also train the in-house SRL model on CoNLL 2012 training data only. When evaluated on the SRL test set, the F1-score for the SRL-data-only, SRL-data+labeled-SNLI, and SRL-data+labeled-MNLI training are 88\%, 76.39\% and 78.89\% respectively. Given that SRL models perform similarly over SNLI and MNLI, SRL quality is unlikely to be the reason for the difference. Therefore, we refute both hypotheses and we plan to explore this direction further in the future.

6 Related Work

It has been acknowledged that though neural models can surpass human performance on large benchmark datasets, they are still far from actually understanding the language. This is true for NLI (Naik et al., 2018; Talman and Chatzikyriakidis, 2019; Gururangan et al., 2018; Poliak et al., 2018; Tsuchiya, 2018; Geva et al., 2019; Nie et al., 2020) and other tasks like question answering (Agrawal et al., 2016; Mudrakarta et al., 2018), reading comprehension (Kaushik and Lipton, 2018), commonsense reasoning (Branco et al., 2021) etc.

SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018) are two large benchmark datasets which motivate the application of neural models developed for the natural language inference task, but the performance of neural models rapidly reach the ceiling level on these two datasets. Adversarial NLI (Nie et al., 2020) was developed with a never-end learning scenario in response to the rapid model improvement. Various probing datasets have been developed to probe about whether the model has learned certain aspects of information humans usually resort to for conducting the same task. For example, Breaking NLI (Glockner et al., 2018) probes about whether the model can distinguish lexical meaning and world knowledge. HANS (McCoy et al., 2019) was created by manipulating the syntactic structure of the premise and the hypothesis to investigate whether the model relies on heuristics or real linguistic understanding. IMPRES (Jeretic et al., 2020) was developed to investigate whether the model has learned pragmatic inferences like presupposition and implicature.

This paper is closely related to work on incorporating explicit linguistic or external knowledge to contextual word representations (Zhang et al., 2019b; Cengiz and Yuret, 2020; Peinelt et al., 2020; Wang et al., 2019b; Kapanipathi et al., 2020) and examining whether the model has learned human-like competence (Goldberg, 2019; Liu et al., 2019a; Hewitt and Manning, 2019; Clark et al., 2019; Lin et al., 2019; Tenney et al., 2019; Warstadt and Bowman, 2020; Manning et al., 2020; Ettinger, 2020; Michael et al., 2020). We propose to improve SemBERT (Zhang et al., 2019b) by predicate-wise concatenation of SRL representations with BERT representations followed with interaction, and analyze whether the infused SRL information increased the model’s performance and linguistic awareness.

7 Conclusion

We conduct experiments to analyze whether incorporating SRL representations into BERT representations by concatenation can improve the models’ performance on the NLI task, and help the models learn linguistic knowledge and human-like generalizations. In addition, we propose predicate-wise concatenation with interaction for combining SRL embeddings and BERT representations, which turns out to be more effective than the sentence-wise concatenation adopted in SemBERT. Through experiments, we observe that infusing SRL representations with BERT representations improves the model generalization on lexical and world knowledge, as evident from the consistently better performance of LingBERT on the Breaking NLI dataset. Performance improvement on HANS lexical overlap heuristic examples when the model is finetuned on SNLI further strengthens our claim. However, we do not observe any semantic-aware model finetuned on MNLI outperform BERT on the HANS dataset. This indicates that incorporating linguistic knowledge explicitly encoded by predicate-argument structures by just concatenating SRL representations with BERT representations does not increase the model’s awareness of other linguistic knowledge and competence of reasoning generalization. Therefore, how to effectively infuse SRL information to improve the model’s awareness of other linguistic phenomena remains an open challenge. With the current model, the extra explicit SRL information does not necessarily reduce the model’s vulnerability to heuristics, and its performance is influenced by the training data.

\textsuperscript{12}Since CoNLL 2012 SRL dataset contains multiple genres, we combined the data for training in order to alleviate the genre difference between SNLI and MNLI and make the comparison more reliable.
References


A Detailed results on HANS dataset

<table>
<thead>
<tr>
<th></th>
<th>train on SNLI</th>
<th></th>
<th>train on MNLI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>all heuristics</td>
<td></td>
<td>all heuristics</td>
</tr>
<tr>
<td></td>
<td>entail</td>
<td>non-entail</td>
<td>overall</td>
</tr>
<tr>
<td></td>
<td>98.85</td>
<td>18.81</td>
<td>58.83</td>
</tr>
<tr>
<td></td>
<td>99.40</td>
<td>16.38</td>
<td>57.89</td>
</tr>
<tr>
<td></td>
<td><strong>99.45</strong></td>
<td><strong>20.48</strong></td>
<td><strong>59.96</strong></td>
</tr>
<tr>
<td>BERT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SemBERT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lexical overlap heuristic</td>
<td>entail</td>
<td>non-entail</td>
<td>overall</td>
</tr>
<tr>
<td>BERT</td>
<td>97.35</td>
<td>46.33</td>
<td>71.84</td>
</tr>
<tr>
<td>SemBERT</td>
<td><strong>98.61</strong></td>
<td>43.02</td>
<td>70.82</td>
</tr>
<tr>
<td>Ours</td>
<td>98.42</td>
<td><strong>54.40</strong></td>
<td><strong>76.41</strong></td>
</tr>
<tr>
<td>subsequence heuristic</td>
<td>entail</td>
<td>non-entail</td>
<td>overall</td>
</tr>
<tr>
<td>BERT</td>
<td>99.67</td>
<td>4.92</td>
<td><strong>52.29</strong></td>
</tr>
<tr>
<td>SemBERT</td>
<td>99.67</td>
<td>3.69</td>
<td>51.68</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>99.96</strong></td>
<td>4.01</td>
<td>51.99</td>
</tr>
<tr>
<td>constituent heuristic</td>
<td>entail</td>
<td>non-entail</td>
<td>overall</td>
</tr>
<tr>
<td>BERT</td>
<td>99.53</td>
<td>5.20</td>
<td><strong>52.36</strong></td>
</tr>
<tr>
<td>SemBERT</td>
<td>99.90</td>
<td>2.44</td>
<td>51.17</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>99.96</strong></td>
<td>3.03</td>
<td>51.49</td>
</tr>
</tbody>
</table>

Table 6: Performance on HANS. *entail* columns are where the ground-truth label is *entailment*, *non-entail* columns are where the ground-truth label is *non-entailment*, and *overall* columns are for both ground-truth labels together.