Joint Training with Semantic Role Labeling for Better Generalization in Natural Language Inference

Cemil Cengiz **Deniz Yuret**
KUIS AI Lab
Koç University, İstanbul, Turkey
ccengiz17,dyuret@ku.edu.tr

Abstract

End-to-end models trained on natural language inference (NLI) datasets show low generalization on out-of-distribution evaluation sets. The models tend to learn shallow heuristics due to dataset biases. The performance decreases dramatically on diagnostic sets measuring compositionality or robustness against simple heuristics. Existing solutions for this problem employ dataset augmentation which has the drawbacks of being applicable to only a limited set of adversaries and at worst hurting the model performance on other adversaries not included in the augmentation set. Our proposed solution is to improve sentence understanding (hence out-of-distribution generalization) with joint learning of explicit semantics. We show that a BERT based model trained jointly on English semantic role labeling (SRL) and NLI achieves significantly higher performance on external evaluation sets measuring generalization performance.

1 Introduction

NLI is the task of determining the inference relationship between a premise and a hypothesis sentence which is usually formulated as a three-class classification task with *entailment*, *contradiction* and *neutral* labels. It has been regarded as a central problem in natural language understanding and found its place in benchmarks such as GLUE (Wang et al., 2018). Contemporary neural network based models achieve state-of-the-art (SOTA) results on these benchmarks. However, scoring high on test sets that have a similar distribution to the training sets does not guarantee wider generalization. Models that top the leaderboards on standard test sets may perform poorly on specifically constructed evaluation sets targeting dataset biases. For instance, the HANS challenge dataset (McCoy et al., 2019b) showed that models trained on NLI get fooled easily by heuristics when the input sentence pairs have high lexical similarity. The following example from HANS can explain how this might happen.

Premise: *The judge by the actor stopped the banker.*

Hypothesis: *The banker stopped the actor.*

A human reading this sentence pair carefully can conclude that the hypothesis can not be inferred from the premise. However, models relying on the lexical overlap heuristic will be fooled and predict the label as *entailment* since the premise contains all words of the hypothesis. Existing approaches commonly tackle such adversaries by training the model with a dataset augmented with similar adversarial examples. As detailed in Section 5, the problem with this approach is that it might lead to overfitting to the adversaries on the augmentation set. Therefore, it can decrease the performance on other possible adversaries and hurt generalization (Nie et al., 2018).

Semantic Role Labeling (SRL) asks the *"who did what to whom, when and where etc."* questions to find the semantic roles of words or phrases in a sentence (He et al., 2017). We hypothesize that using the SRL task as a joint objective should improve the semantic knowledge of the models, thus making them less prone to dataset biases. Consider the semantic roles in the previous example sentence pair, which are shown in Table 1. We can see that the role of *"the banker"* differs between the sentences. Since *"stop"* is not a reciprocal verb, a model that is aware of the semantic roles can find out that the inference relation is *non-entailment* although the premise contains all words in the hypothesis, albeit with a different order. In contrast, if a model pays too much attention to lexical similarity, it might falsely predict the relation as *entailment* as the SOTA models analyzed on HANS such as BERT (Devlin et al., 2018) do. SRL informs the model directly about the semantic roles of words

Proceedings of the 5th Workshop on Representation Learning for NLP (RepL4NLP-2020), pages 78–88
July 9, 2020.

Premise	The judge by the actor stopped the banker.
VERB	stopped
ARG0	The judge by the actor
ARG1	the banker

Hypoth.	The banker stopped the actor.
VERB	stopped
ARG0	The banker
ARG1	the actor

Table 1: An example pair from HANS, including the semantic roles of the words in each sentence. *ARG0* represents the proto-agent, i.e. the thing that stops, *ARG1* represents the proto-patient, i.e. the object being stopped, in this example.

which makes it easier for the model to rely on more than just shallow lexical cues.

Our contribution in this work is threefold:

- We propose a BERT based multi-task learning (MTL) model jointly trained on English SRL and NLI (Section 3), and show that this model achieves scores comparable to the single-task BERT on both tasks (Section 4).

- We evaluate the proposed model on out-of-distribution test sets such as HANS (McCoy et al., 2019b) and Comparisons (Dasgupta et al., 2018) and demonstrate that it exceeds the single-task BERT performance significantly. Specifically, when trained on MultiNLI, the multi-task model exceeds the single-task model accuracy by 4% on HANS and 5.3% on Comparisons without using data augmentation. (Section 4.1.2 and 4.2)

- We compare the proposed MTL approach to the sequential transfer learning and show that the MTL is more helpful. (Section 4.1.3)

2 Datasets

In this section, we describe the datasets used in the experiments. We explain the shortcomings of the NLI datasets and describe an SRL dataset that can be used to alleviate these shortcomings.

2.1 Large-scale NLI Datasets

In this work, we consider two open-domain, large-scale NLI datasets in English, SNLI (Bowman et al., 2015) and MultiNLI (Williams et al., 2018). SNLI is created using image captions written by humans whereas MultiNLI includes five different genres of written and spoken English such as travel guides and telephone conversations. Both datasets have been used for training general NLI models, or as an intermediate training resource for transfer learning to a domain-specific dataset, possibly with smaller size (Cengiz et al., 2019). Recently, deep neural network models achieved human-level performance on NLI tasks in benchmarks such as GLUE (Wang et al., 2018) and SuperGLUE (Wang et al., 2019). However, as McCoy et al. (2019b) and Dasgupta et al. (2018) showed, these results do not reflect the performance of the models on out-of-distribution test sets. They proposed such adversarial test sets targeting specific biases apparent in the original NLI datasets to show that SOTA models are vulnerable to these superficial patterns.

2.2 Adversarial NLI Datasets

HANS is an extensive evaluation set proposed by McCoy et al. (2019b), that consists of three types of adversarial examples: *lexical overlap*, *subsequence* and *constituent*. *Lexical overlap* is the most general category, indicating all the words in the hypothesis sentence are present in the premise as well. An example from this category with *entailment* gold label is the following: *"The banker near the judge saw the actor. ⇒ The banker saw the actor." Subsequence* is a special case of *lexical overlap*, indicating that the hypothesis is a contiguous subsequence of the premise. The following pair is an example from this category with *entailment* gold label: *"The artist and the student called the judge. ⇒ The student called the judge."* Lastly, *constituent* is a special case of the *subsequence*, denoting that the hypothesis is a complete subtree of the premise's constituency parse tree. An instance from *constituent* category with *non-entailment* gold label is as following: *"If the actor slept, the judge saw the artist. ⇒ The actor slept."* It is worth noting that although there is a hierarchical relation between the categories, their instances do not overlap. Therefore, they will be treated as distinct categories throughout the paper. The sentences included in this dataset were created using templates and ensured to be plausible. Moreover, the verbs were chosen from the frequently used verbs in MultiNLI so that the models trained on MultiNLI are familiar with them. Differently than MultiNLI, this dataset has binary labels, a label is either *entailment* or *non-entailment*. Finally, this dataset has two parti-

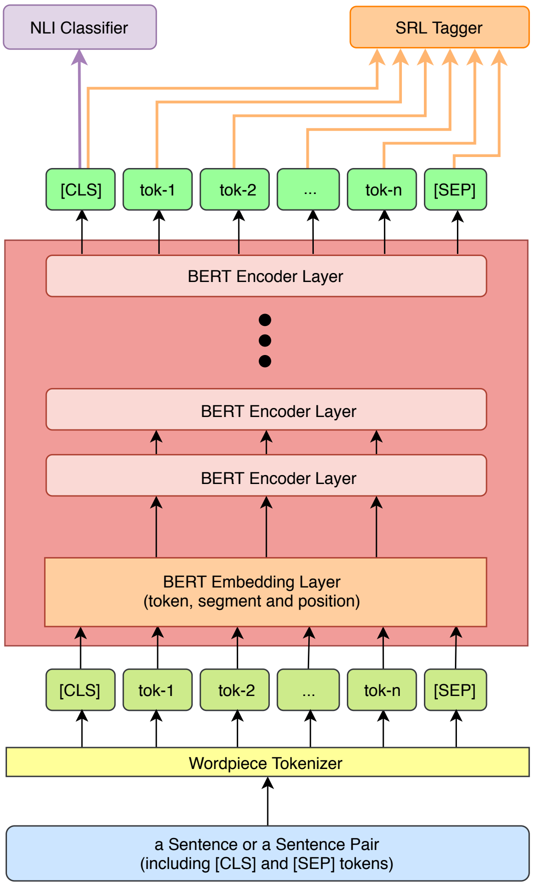

Figure 1: Multi-task BERT model for SRL and NLI. Each task-specific head contains a linear layer to transform the embeddings into the task's label space.

tions with identical size, a test set to evaluate the models and an augmentation set to augment the MultiNLI training set. In our experiments, we treat the augmentation set as a validation set and do not use it for training.

The Comparisons dataset (Dasgupta et al., 2018) attempts to evaluate the models trained on SNLI for three types of adversarial examples: *same*, *more-less*, *not*. The *same* category consists of hypothesis-premise pairs having exactly the same words in a different order. An example with *contradiction* gold label is as following: *"The woman is more cheerful than the man. ⇒ The man is more cheerful than the woman."* The *more-less* type contains instances whose sentence pair differ by including the word *"more"* or the word *"less"*, and possibly in word order. The following pair is an example from this category with *entailment* label: *"The woman is more cheerful than the man. ⇒ The man is less cheerful than the woman."* The third category is the *not* type, representing the instances having the negation word *"not"* either in the hypothesis or in the premise, but not in both. The word order of the sentences might differ as well. A sentence pair from this category with *entailment* gold label is the following: *"The woman is more cheerful than the man. ⇒ The man is not more cheerful than the woman."* The authors created this dataset by automatically generating examples fitting into one of the described categories using a vocabulary similar to SNLI's. Moreover, they analyzed SNLI and showed that it has many examples fitting into the examined categories with labels mostly supporting the heuristic choice. Unlike SNLI, this dataset does not have the *neutral* label. It has the *entailment* label for the positive examples and the *contradiction* label for the negative examples.

2.3 Semantic Role Labeling as an Auxiliary Objective for Sentence Understanding

In this work, we use the English Ontonotes v5.0 SRL dataset with the CONLL-2012 shared task format (Pradhan et al., 2013) which gives the predicate-argument structure for each sentence. The auxiliary task we used is formulated as prediction of the arguments for a given predicate in a sentence. Therefore, each predicate in a sentence together with the semantic role label spans associated with it yield a different training instance. The number of training instances in the whole dataset is around 280,000.

3 Model Description

We propose a multi-task BERT model to jointly predict semantic roles and perform natural language inference. BERT is used as the shared encoder module and two separate decoder heads are appended on top of it to perform task-specific operations. The overall picture of the model can be seen from Figure 1. Following Liu et al. (2019), the tasks share the encoder part of the model including the lexicon encoder and all BERT layers.

We follow the original sentence pair classification formulation for BERT while training it for NLI task. On the input side, we concatenate the premise and hypothesis tokens, add a [SEP] token at the end of both sentences, and finally add a [CLS] token at the beginning of the whole sequence. Figure 2 shows the token embedding for an example sentence pair. While processing an NLI input, we take the [CLS] token embedding at the BERT's output and treat it as the summary of the whole sequence. The dimension of this embedding is reduced to

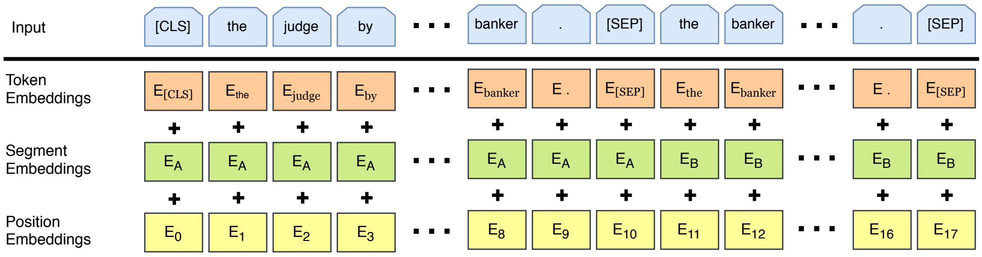

Figure 2: Input representation of *"The judge by the actor stopped the banker. ⇒ The banker stopped the actor."* sentence pair for NLI task.

three after passing through a two-layer MLP since there are three labels in MultiNLI and SNLI. Finally, a softmax is applied to get the probability for each label class. Following McCoy et al. (2019b), when we evaluate the model on HANS or Comparisons, we collapse the predicted *contradiction* and *neutral* labels into a single negative label to output binary labels.

For the SRL training, we adapt an architecture similar to the one proposed by Shi and Lin (2019). In our implementation, we indicate the predicate using the segment embeddings by assigning 1 to the predicate word pieces and 0 to the other tokens. Figure 3 denotes the embeddings used for the SRL. We opt for a simple decoder and rely purely on the self attention to capture contextual information. The embedding of each token is directly passed through a two-layer MLP independently, and the SRL tag is determined by a final softmax layer. We use the Inside Outside Beginning (IOB)[1] tagging for spans, and a Viterbi decoder to ensure prediction of valid spans during testing. As Figure 2 and 3 show, the segment embeddings represent different things for SRL and NLI inputs. In NLI, segment embeddings separate the sentences whereas they indicate the target verb in SRL. Moreover, how the BERT outputs are processed is also different between the tasks. In spite of these differences, the training is expected to optimize the BERT weights so that it can generate embeddings suitable for both tasks. Intuitively, this representation will be less prone to dataset biases in the NLI thanks to explicitly forcing the model to pay attention to the semantic roles of the words.

4 Experiments and Results

In this section, we present the experiments we conducted by training the BERT based model in single-task and multi-task learning setups. MultiNLI and SNLI datasets are used to train the models whereas HANS and Comparisons are used for evaluation. For the multi-task learning experiments, we used the SRL dataset for joint training with NLI. In both of the HANS and Comparisons experiments, we tuned the hyperparameters using a validation set from the same distribution as the adversarial test set. Nevertheless, we also tested our highest performing models on the original MultiNLI and SNLI validation sets to make sure our multi-task BERT model performs well on those as well. Indeed, the multi-task model performance on the original validation sets turned out to be similar (accuracy difference is around $\pm 0.5\%$, depending on the hyperparameters) to the single-task BERT performance.

[1] In IOB style tagging, each span begins with a *B-* tag and continues with *I-* tags except for *Other* tokens, which takes *O* tags.

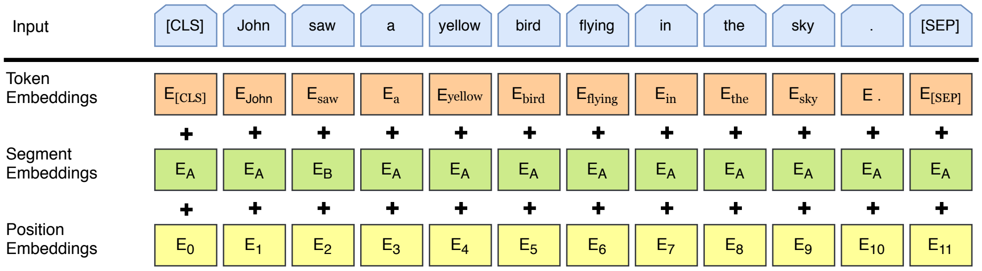

Figure 3: Input representation of *"John saw a yellow bird flying in the sky."* sentence for SRL task when the predicate is *saw*.

4.1 HANS Experiments

We experimented with direct training on NLI, sequential transfer learning using SRL, and a multi-task learning approach to train on the NLI and SRL tasks jointly. As HANS was proposed as an adversarial evaluation set for the NLI models trained on MultiNLI, we use it as the NLI training dataset. Following McCoy et al. (2019b), we output *non-entailment* label when a model predicts *contradiction* or *neutral*.

4.1.1 Single-task MultiNLI Training

We trained a single-task BERT model on the MultiNLI dataset to be used as the baseline for the HANS evaluation. The BERT weights are initialized with the pre-trained weights from Devlin et al. (2018) whereas the classifier head is randomly initialized. During training, all weights are updated. We used the HANS augmentation dataset as the development set for hyperparameter tuning.

As reported by McCoy et al. (2019b), BERT performs poorly on HANS although better than bag-of-words or LSTM (Hochreiter and Schmidhuber, 1997) based models. However, the follow-up work by McCoy et al. (2019a) showed BERT's performance on HANS varied dramatically depending on the order of instances fed during training and the initial weights of the classifier head, both of which can be varied by changing the random seeds. To further investigate that, they repeated the training of BERT 100 times with the same settings except that randomness and compared the results. The largest variance was encountered on the *lexical overlap* category when the gold label is *non-entailment* whereas the other results were close among different runs. In our experiment, we got a 51% accuracy on *lexical overlap*, which is close to the upper limit of the range (6% – 54%) reported by McCoy et al. (2019a). Table 2 includes the comparison of the original results (McCoy et al., 2019b) with our run. Our results are comparable so we use this model as the baseline for single-task BERT.

4.1.2 Multi-task Training for SRL and MultiNLI

In this part, we present the result of training BERT on SRL and MultiNLI jointly with the multi-task approach described in Section 3. We used the HANS augmentation dataset as the validation set for MultiNLI, and CoNLL-2012 development set for SRL validation. We validated the trained model against both validation set separately at the end of each epoch. Then, the model performed highest on the HANS augmentation set was evaluated on the HANS test set. The results are shown in Table 2, together with the single-task results for comparison. The multi-task approach improved the overall accuracy by 4%. Although there is a slight decrease in the results when the correct label is *entailment*, this is an expected drop. The accuracy of the single-task model reaches 100% on the *subsequence* category, and is at least 96% on the remaining two categories when the correct label is *entailment*. Since the MultiNLI instances involving a heuristic examined by HANS are mostly labeled with *entailment*, the models tend to assign *entailment* labels to such examples in the HANS dataset. Because our multi-task model is less severely affected by the heuristics, it is less likely to output *entailment* when these heuristics are encountered. As one would expect, the gains come from the instances whose correct label is *non-entailment*. Noticeably, the multi-task training improved the accuracy for this label in all three categories dramatically.

To examine the improvements in more detail, we present the results broken down into subcategories in Table 3. We refer the readers to McCoy

	Correct: Entailment			**Correct: Non-entailment**			
BERT model	**Lexical**	**Subseq.**	**Const.**	**Lexical**	**Subseq.**	**Const.**	**Avg.**
McCoy et al. (2019b)	0.95	0.99	**1.00**	0.16	0.04	0.16	0.55
Single-task	**0.96**	**1.00**	0.99	0.51	0.05	0.18	0.62
Multi-task	0.91	0.98	0.95	**0.71**	**0.13**	**0.25**	**0.66**

Table 2: Comparison of the previous work (McCoy et al., 2019b) and our single-task and multi-task BERT models on HANS. All models started from pre-trained weights. The multi-task model was jointly trained on SRL and MultiNLI whereas single-task models were only trained on MultiNLI. The highest accuracy for each category is indicated with bold. Note that (McCoy et al., 2019b) results on the *entailment* and *non-entailment* categories were obtained by averaging the subcases using the BERT column of Table 7 and Table 8 respectively in their paper. (**Lexical**: lexical overlap, **Subseq.**: subsequence, **Const.**: constituent)

Category	Single-Task	Multi-Task
Lexical Overlap		
Subject-object swap	0.68	0.83
Preposition	0.68	0.79
Relative clause	0.60	0.73
Conjunction	0.55	0.70
Passive	0.01	0.51
Subsequence		
NP/S ambiguity	0.00	0.03
Prepositional phrase on subject	0.14	0.20
Relative clause on subject	0.10	0.24
Past participle	0.00	0.06
NP/Z ambiguity	0.02	0.15
Constituent		
Embedded under if	0.46	0.71
After if clause	0.00	0.00
Embedded under verb	0.31	0.47
Disjunction	0.07	0.02
Adverb	0.08	0.06

Table 3: Fine-grained comparison of single-task and multi-task BERT on HANS's three different categories when the correct label is *non-entailment*. The rows denote the heuristic type found in the sentences.

et al. (2019b) for details and examples of the subcategories. The top part of the Table 3 shows the fine-grained results for the *lexical overlap* category with the *non-entailment* gold labels. Although all subcategories improved, the largest gain comes from the *passive* examples. The *passive* case with *non-entailment* labels includes examples with sentence pairs almost identical to each other, only one of them has an active verb while the other has the passive form of it. An example sentence pair is the following: *"The senators were helped by the managers. ⇒ The senators helped the managers."* The single-task model misclassified almost all *non-entailment* examples involving passive sentences whereas the multi-task model could predict them correctly half of the time. This is a significant improvement, the multi-task model has begun to identify the meaning changes when a verb is switched from passive to active while the word order is kept unchanged.

The middle section of Table 3 shows the results on the *subsequence* category for the *non-entailment* gold labels. All subcategories improved although the degree is less than *lexical overlap*. The largest improvement is found on *relative clause on subject*, which represents the sentence pairs differing by their subjects such that the premise's subject is a relative clause and the hypothesis's subject is a particular segment of that clause that leads to *non-entailment*. An example sentence pair from this category is the following: *"The secretary that admired the senator saw the actor. ⇒ The senator saw the actor."*. When trained with a multi-task approach, the model makes some progress on recognizing that the overall meaning of a relative clause does not necessarily entail a part of it.

The last category we investigated is *constituent*s with *non-entailment* gold labels, whose results are given in the bottom part of Table 3. There are significant improvements in two subcategories whereas the remaining three did not improve or very slightly dropped. The largest improvement (25% accuracy) is on *embedded under if* subcategory, which denotes the examples with a premise having an *if* (or *unless*) clause whereas the hypothesis has the result part of the *if* clause. An example from this subcategory is: *"Unless the authors saw the students, the doctors resigned. ⇒ The doctors resigned."* The second largest gain (16% accuracy) comes from *embedded under verb* subcategory, which is similar to the previous one, except that the embedding is achieved using a verb. An example is: *"The tourists said that the lawyer saw the banker. ⇒ The lawyer saw the banker."*.

As shown by the HANS results, there are solid improvements on NLI evaluation after switching to the multi-task training. However one needs to ask if the joint training hurts the other task, SRL. In the multi-task experiment, the F1 score on SRL test dataset is 86.0 which is comparable to the single model SOTA results noted by Shi and Lin (2019). Therefore, the joint training did not harm the SRL performance, while improving the out-of-distribution performance on NLI.

4.1.3 Sequential Transfer Learning from SRL to MultiNLI

We experimented with sequential transfer learning to test if a simple transfer learning strategy is

Model	same	more /less	not	Avg.
BOW-MLP	50.0	50.0	49.9	50.0
InferSent	51.4	**50.1**	47.8	49.8
BERT	**85.3**	47.9	44.5	59.2
MTL-BERT	80.5	47.9	**51.3**	**59.9**

Table 4: Percent accuracy of the models on Comparisons dataset. BERT based models are our implementations while the others are from Dasgupta et al. (2018). Multi-task (MTL) BERT is trained on SRL and SNLI. The highest accuracy for each category is indicated with bold. Note that the BOW-MLP and InferSent rows are obtained by merging the *neutral* and *contradiction* labels in Figure 2 and 3 from Dasgupta et al. (2018).

enough to carry information from the SRL task so that the model is more robust to the biases in the NLI dataset. First, an SRL tagger head with random weights is appended on top of the pre-trained BERT encoder. This model is fine-tuned on SRL until the F1 score on the SRL validation set is maximized. The model weights from the epoch resulting in the highest SRL development set score is stored. Then, its SRL head is stripped, and an NLI classifier head with random weights is appended on top of the [CLS] token. Finally, the model is trained on MultiNLI and validated against HANS augmentation set. After training, the model is evaluated on HANS test set. The result is within the accuracy range for the single-task training results reported by McCoy et al. (2019a). This shows that our transfer learning strategy did not improve HANS results over the BERT trained only on MultiNLI. We anticipate that this is because the model forgets most of the knowledge about the SRL task during NLI training. To avoid that, we switched to multi-task setup presented in Section 3 to learn SRL and NLI jointly so that the semantic role knowledge is not forgotten.

4.2 Comparisons Dataset Results

We trained BERT with single-task and multi-task learning approaches and compared them on the Comparisons dataset. First, we used the SNLI dataset as the NLI training source following Dasgupta et al. (2018). We used the validation set released with the Comparisons dataset for hyperparameter optimization during training of both the single-task and multi-task models. Unlike SNLI, this dataset contains only two labels, *entailment* and *contradiction*. Therefore, differently than Dasgupta et al. (2018), we converted the predicted *neutral* labels to *contradiction* to have a unified negative label. Table 4 compares the overall performance of our BERT based models and the previously examined models on the test set. InferSent (Conneau et al., 2017) is a sentence encoding based NLI model that uses LSTM as the encoder. Although it is more complex than the bag-of-words (BOW-MLP) model, their performances are similar on this set. We see that the performance of BERT models on the *same* category are much better than the simpler models. The high performance of BERT models on this category can be attributed to the fact that the BERT was pre-trained on a large corpus with missing word prediction and next sentence prediction tasks, making it more aware of the word order. However, in the remaining two categories, both the single-task and multi-task BERT perform relatively close to the remaining models.

MultiNLI is a more diverse dataset compared to SNLI, including examples from several genres. Therefore, we also tried MultiNLI as the NLI training source and replicated the experiments to see how the single-task and multi-task BERT performance will change. The results are given in Table 5 together with the SNLI based results for comparison. We see that switching to MultiNLI improved both models substantially. However, the increase in the multi-task model is significantly more prominent, showing the advantage of the joint training with SRL. The multi-task model correctly classifies almost all test examples in the *more/less* category and most of the *not* category. However, the trend of observing better performance on the *same* category from the single-task model holds here as well. This result is surprising and needs further investigation.

	Training set: SNLI				Training set: MultiNLI			
BERT Model	same	more/less	not	Avg.	same	more/less	not	Avg.
Single-task	85.3	47.9	44.5	59.2	74.1	88.3	74.3	78.9
Multi-task	80.5	47.9	51.3	59.9	63.3	97.3	91.9	84.2

Table 5: Percent accuracy of the BERT models on Comparisons dataset.

4.3 Training Details

We used the PyTorch (Paszke et al., 2017) framework and the AllenNLP (Gardner et al., 2018) library for implementation. We adapted some code to implement the multi-task training logic from Sanh et al. (2019)'s hierarchical multi-task learning project[2]. In all experiments, we used the base version of BERT by initializing it with the weights released by Devlin et al. (2018).

We use uniform mini-batches, i.e. a mini-batch contains instances from a single-task. Each dataset is divided into mini-batches with the same size and an iterator for each of them is created that can cycle through a dataset and provide batches indefinitely. In a training step, we decide which task to train with a probabilistic sampling, get a mini-batch from the iterator for that task, and perform a forward pass on it and back-propagate the loss. During the back-propagation, we update the task-specific head of the chosen task, as well as the BERT encoder. Following Sanh et al. (2019), we use proportional sampling to decide on the task type at the beginning of each training step.

Recent studies generally use a single global optimizer for all tasks (Sanh et al., 2019; Liu et al., 2019). In this work, we tried both this approach and using a different optimizer for each task. The advantage of using multiple optimizers is that the learning rates of the individual tasks can be set to different values, and each task can have its own learning rate scheduler. We used *BertAdam* optimizer from HuggingFace, and set its maximum learning rate to 2e-5 or 5e-5 according to the validation accuracy on the NLI evaluation task. Moreover, we employed a slanted triangular learning rate scheduler (Howard and Ruder, 2018) with a cut fraction of 0.1 and decay factor of 0.38. In all experiments the maximum sequence length is set to 256, and longer sequences are truncated. In all training experiments, 4 GPUs were used in parallel and the datasets were divided into mini-batches of size 12 based on GPU memory limitations.

5 Related Work

In this section, we discuss various solution approaches proposed for the NLI task. We start with a sentence embedding based approach and continue with a data augmentation method targeting the generalization problem of NLI models. Then, we discuss some models benefiting from syntax or semantic roles and touch on multi-task models.

Some previous studies used sentence embedding based approaches to solve NLI. Noticeably, InferSent (Conneau et al., 2017) uses an LSTM to encode the premise and hypothesis sentences independently. Then, it concatenates the premise, hypothesis embeddings and two feature vectors obtained by their element-wise multiplication and absolute difference to get the overall sentence pair representation. Finally, an MLP layer followed by a softmax is used to calculate the class scores. Being trained on SNLI, this model suffers from the biases in its training data and performed close to the BOW model on the Comparisons evaluation set.

There are a number of studies that use data augmentation to address the generalization problem revealed by NLI challenge datasets like HANS and Comparisons. McCoy et al. (2019b); Dasgupta et al. (2018); Nie et al. (2018) created augmentation sets consisting of training instances with properties similar to the proposed adversarial evaluation set. The augmentation set is focused on the examined phenomena and considerably smaller than the original training set in general. Nevertheless, the models are shown to achieve very strong results, even close to $\%100$ accuracy on the evaluation sets after training with augmented datasets. However, there are some problems with an augmentation approach performed this way, i.e. using a new dataset targeting the inspected phenomena in the proposed evaluation set. First of all, it is not clear if they do result in improvement on the language understanding of the model in general. Rather, the model at hand is patched so that it can excel on some specific cases that the new evaluation dataset examines. However, one can presumably find other adversarial example classes for a given training dataset, so creating an augmentation set for each possible adversarial class may not be feasible. Moreover, Nie et al. (2018) showed that augmenting the training dataset by targeting some specific category of adversarial examples might be harmful to other types of adversarial examples. In other words, dataset augmentation with such limited focus might lead to overfitting to the targeted adversaries and hurt the overall robustness. Therefore, in this work, we took a different approach and introduced an inductive bias on the model by explicitly enforcing it to produce representations suitable to extract semantic

[2] https://github.com/huggingface/hmtl

role information.

Some recent studies have investigated the benefits of semantic role labeling on the performance of natural language inference models. Noticeably, Zhang et al. (2020, 2018) used SRL as a supplementary task for text comprehension tasks such as textual entailment and question answering. Similar to our work, they used PropBank (Palmer et al., 2005) style role annotations and treated SRL as a sequence tagging problem. Zhang et al. (2020)'s approach is different from ours in that they use a pre-trained, SOTA SRL model to provide semantic embeddings to enrich the contextual embeddings from BERT. They kept the SRL model frozen and trained other parts of the model including the BERT encoder. Similarly, Zhang et al. (2018) employed two different networks where one of them is an SRL model responsible for generating the semantic embeddings to support the other network which is trained to solve the downstream task at hand. Moreover, both networks in this model use pre-trained word embeddings such as ELMo (Peters et al., 2018) or GloVe (Pennington et al., 2014). The main difference of our approach from these is that we use a single network and train it in a multi-task fashion by sharing encoder representations among different tasks. Moreover, unlike our work, they evaluated their models on the original datasets e.g. MultiNLI, so their focus was not to improve the model performance on the adversarial evaluation sets.

Instead of semantic role information, some recent works investigated the benefits of syntax to support the natural language inference models. Noticeably, Pang et al. (2019) used the hidden word representations of an externally trained, high performing dependency parser to enrich the BERT based NLI models. With this approach, the models achieved some modest increase on the overall HANS results.

Multi-task models powered with pre-trained language model based encoders have achieved SOTA performance on natural language understanding benchmarks such as GLUE (Wang et al., 2018), and SuperGLUE (Wang et al., 2019). However, there is not much work focusing on simultaneously solving both a word-level semantics task such as SRL and a sentence-level understanding such as NLI which requires higher level reasoning. Instead, the existing approaches such as Liu et al. (2019); Clark et al. (2019) combine multiple sentence-level understanding tasks to solve them jointly without any aid from lower level tasks focusing on syntax or word-level semantics. Our approach differs from them in that we hypothesize using both word-level semantics and high level reasoning tasks might be a more suitable approach to learn deeper understanding of the sentences, thereby suffering less from the dataset biases in reasoning tasks such as NLI.

Some previous work attempted to cast NLI to a different natural language understanding task such as question answering. Particularly, McCann et al. (2018) suggested a collection of various tasks including NLI for a benchmark and proposed a novel approach to solve all those tasks using a single multi-task model. They casted each task to the question answering problem and trained a model to solve all of them jointly.

6 Conclusion and Future Work

This work presents a multi-task learning approach using SRL task to apply an inductive bias on a BERT based NLI model. Our experiments show that joint training with SRL makes the model more robust to the superficial patterns in the NLI training data. As opposed to the augmentation based solutions focused on specific adversarial classes, this approach has the advantage of being applicable to a variety of adversaries without overfitting to some of them. Having access to the semantic role information improves the sentence understanding of the model, hence making it generalize better to the unseen dataset distributions including the adversarial ones such as HANS and Comparisons. The SRL task utilized in this work processes a single predicate per data instance. The future work might incorporate joint prediction of all predicates and corresponding roles to analyze its effect on adversarial NLI evaluation performance.

Acknowledgments

The authors would like to thank Ulaş Sert and Ceyda Özler for their help in creating the figures and the anonymous reviewers for their valuable feedback. Cemil Cengiz is supported by Huawei Turkey R&D Center through the Huawei Graduate Research Support Scholarship.

References

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

In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Cemil Cengiz, Ulaş Sert, and Deniz Yuret. 2019. KU_ai at MEDIQA 2019: Domain-specific pre-training and transfer learning for medical NLI. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 427–436, Florence, Italy. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D. Manning, and Quoc V. Le. 2019. BAM! born-again multi-task networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5931–5937, Florence, Italy. Association for Computational Linguistics.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. 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. Association for Computational Linguistics.

Ishita Dasgupta, Demi Guo, Andreas Stuhlmüller, Samuel J Gershman, and Noah D Goodman. 2018. Evaluating compositionality in sentence embeddings. *arXiv preprint arXiv:1802.04302*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 473–483, Vancouver, Canada. Association for Computational Linguistics.

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

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, Florence, Italy. Association for Computational Linguistics.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.

R. Thomas McCoy, Junghyun Min, and Tal Linzen. 2019a. Berts of a feather do not generalize together: Large variability in generalization across models with similar test set performance.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019b. 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. Association for Computational Linguistics.

Yixin Nie, Yicheng Wang, and Mohit Bansal. 2018. Analyzing compositionality-sensitivity of NLI models. *CoRR*, abs/1811.07033.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Deric Pang, Lucy H. Lin, and Noah A. Smith. 2019. Improving natural language inference with a pre-trained parser.

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

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. 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. Association for Computational Linguistics.

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

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152, Sofia, Bulgaria. Association for Computational Linguistics.

Victor Sanh, Thomas Wolf, and Sebastian Ruder. 2019. A hierarchical multi-task approach for learning embeddings from semantic tasks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:6949–6956.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *CoRR*, abs/1904.05255.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *NeurIPS*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. 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. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. 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. Association for Computational Linguistics.

Zhuosheng Zhang, Yuwei Wu, Zuchao Li, and Hai Zhao. 2018. Explicit contextual semantics for text comprehension.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020. Semantics-aware BERT for language understanding. In *the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-2020)*.