Discriminatively-Tuned Generative Classifiers for Robust Natural Language Inference

Xiaoan Ding[1] * **Tianyu Liu**[3] *† **Baobao Chang**[3 4] **Zhifang Sui**[3 4] **Kevin Gimpel**[2]
[1] University of Chicago, IL, USA [2] Toyota Technological Institute at Chicago, IL, USA
[3] Peking University, Beijing, China [4] Peng Cheng Laboratory, Shenzhen, China
xiaoanding@uchicago.edu, {tianyu0421, chbb, szf}@pku.edu.cn,
kgimpel@ttic.edu

Abstract

While discriminative neural network classifiers are generally preferred, recent work has shown advantages of generative classifiers in term of data efficiency and robustness. In this paper, we focus on natural language inference (NLI). We propose GenNLI, a generative classifier for NLI tasks, and empirically characterize its performance by comparing it to five baselines, including discriminative models and large-scale pretrained language representation models like BERT. We explore training objectives for discriminative fine-tuning of our generative classifiers, showing improvements over log loss fine-tuning from prior work (Lewis and Fan, 2019). In particular, we find strong results with a simple unbounded modification to log loss, which we call the "infinilog loss". Our experiments show that GenNLI outperforms both discriminative and pretrained baselines across several challenging NLI experimental settings, including small training sets, imbalanced label distributions, and label noise.

1 Introduction

Natural language inference (NLI) is the task of identifying the relationship between two fragments of text, called the *premise* and the *hypothesis* (Dagan et al., 2005; Dagan et al., 2013). The task was originally defined as binary classification, in which the labels are *entailment* (the premise implies the hypothesis) or *not entailment*. Subsequent variations added a third *contradiction* label. Most models for NLI are trained and evaluated on standard benchmarks (Bowman et al., 2015; Williams et al., 2018; Wang et al., 2018) in a discriminative manner (Conneau et al., 2017; Chen et al., 2017a). These benchmarks typically have relatively clean, balanced, and abundant annotated data, and there is no distribution shift between the training and test sets.

However, when data quality and conditions are not ideal, there is a substantial performance decrease for existing discriminative models, including both simple model architectures and more complex ones. Prior work on document classification and question answering has shown that **generative classifiers** have advantages over their discriminative counterparts in non-ideal conditions (Yogatama et al., 2017; Lewis and Fan, 2019; Ding and Gimpel, 2019).

In this paper, we develop generative classifiers for NLI. Our model, which we call GenNLI, defines the conditional probability of the hypothesis given the premise and the label, parameterizing the distribution using a sequence-to-sequence model with attention (Luong et al., 2015) and a copy mechanism (Gu et al., 2016). We explore training objectives for discriminative fine-tuning of our generative classifiers, comparing several classical discriminative criteria. We find that several losses, including hinge loss and softmax-margin, outperform log loss fine-tuning used in prior work (Lewis and Fan, 2019) while similarly retaining the advantages of generative classifiers. We also find strong results with a simple unbounded modification to log loss, which we call the "infinilog loss".

Our evaluation focuses on challenging experimental conditions: small training sets, imbalanced label distributions, and label noise. We empirically compare GenNLI with several discriminative baselines and large-scale pretrained language representation models (Devlin et al., 2019; Yang et al., 2019; Liu et al., 2019) on five standard datasets. GenNLI has better performance than discriminative classifiers under the small data setting. Moreover, when limited to 100 instances per class, GenNLI consistently outperforms all BERT-style pretrained models on four of the five datasets. These results

*Equal contribution.
†Contribution during visiting TTIC.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 8189–8202,
November 16–20, 2020.

are appealing especially in comparison with BERT-style pretrained baselines. Large-scale pretrained language models have achieved state-of-the-art results on a wide range of NLP tasks, but they still require hundreds or even thousands of annotated examples to outperform GenNLI.

GenNLI also outperforms discriminative classifiers when the training data shows severe label imbalance and when training labels are randomly corrupted. We additionally use GenNLI to generate hypotheses for given premises and labels. While the generations tend to have low diversity due to high lexical overlap with the premise, they are generally fluent and comport with the given labels, even in the small data setting.

2 Background and Related Work

2.1 Generative Classifiers

While discriminative classifiers directly model the posterior probability of the label given the input, i.e., $p(y \mid x)$, generative classifiers instead model the joint probability $p(x, y)$, typically factoring it into $p(x \mid y)$ and $p(y)$ and making decisions as follows:

$$\hat{y} = \underset{y}{\operatorname{argmax}}\, p(x \mid y)p(y)$$

Most neural network classifiers are trained as discriminative classifiers as these work better when conditions are favorable for supervised learning, namely that training data is plentiful and that the training and test data are drawn from the same distribution. While discriminative classifiers are generally preferred in practice, there is certain prior work showing that generative classifiers can have advantages in certain conditions, especially when training data is scarce, noisy, and imbalanced (Yogatama et al., 2017; Lewis and Fan, 2019; Ding and Gimpel, 2019).

Ng and Jordan (2002) proved theoretically that generative classifiers can approach their asymptotic error much faster, as naïve Bayes is faster than its discriminative analogue, logistic regression. Yogatama et al. (2017) compared the performance of generative and discriminative classifiers and showed the advantages of neural generative classifiers in terms of sample complexity, data shift, and zero-shot and continual learning settings. Ding and Gimpel (2019) further improved the performance of generative classifiers on document classification by introducing discrete latent variables into the generative story. Lewis and Fan (2019) developed generative classifiers for question answering and achieved comparable performance to discriminative models on the SQuAD (Rajpurkar et al., 2016) dataset, and much better performance in challenging experimental settings.

In this paper, we develop generative models for natural language inference inspired by models for sequence-to-sequence tasks. We additionally contribute an exploration of several discriminative objectives for fine-tuning our generative classifiers, finding multiple choices to outperform log loss used in prior work. We also compare our generative classifiers with fine-tuning of large-scale pretrained models, and characterize performance under other realistic settings such as imbalanced and noisy datasets.

2.2 Natural Language Inference

Early methods for NLI mainly relied on conventional, feature-based methods trained from small-scale datasets (Dagan et al., 2013; Marelli et al., 2014). The release of larger datasets, such as SNLI, made neural network methods feasible. Such methods can be roughly categorized into two classes: sentence embedding bottleneck methods which first encode the two sentences as vectors and then feed them into a classifier for classification (Conneau et al., 2017; Nie and Bansal, 2017; Choi et al., 2018; Chen et al., 2017b; Wu et al., 2018), and more general methods which usually involve interactions while encoding the two sentences in the pair (Chen et al., 2017a; Gong et al., 2018; Parikh et al., 2016). Recently, NLI models are shown to be biased towards spurious surface patterns in the human annotated datasets (Poliak et al., 2018; Gururangan et al., 2018; Liu et al., 2020a), which makes them vulnerable to adversarial attacks (Glockner et al., 2018; Minervini and Riedel, 2018; McCoy et al., 2019; Liu et al., 2020b).

3 A Generative Classifier for NLI

Each example in a natural language inference dataset consists of two natural language texts, known as the premise and the hypothesis, and a label indicating the relation between the two texts. Formally, we denote an instance $\langle x^{(p)}, x^{(h)}, y \rangle$ as a tuple consisting of a premise $x^{(p)} = \{x_1^{(p)}, x_2^{(p)}, ..., x_N^{(p)}\}$, a hypothesis $x^{(h)} = \{x_1^{(h)}, x_2^{(h)}, ..., x_T^{(h)}\}$, and a label $y \in Y$.

Most existing NLI models are trained in a dis-

criminative manner by maximizing the conditional log-likelihood of the label given the input, i.e., $\log p(y \mid x^{(p)}, x^{(h)})$. In this paper, we propose generative classifiers for NLI that are trained instead to estimate the probability of the hypothesis given the premise and the label, i.e., $p(x^{(h)} \mid x^{(p)}, y)$, typically by maximizing log-likelihood. We decompose this conditional probability using the chain rule, and our final training objective is to minimize the following negative log likelihood:

$$L(x^{(p)}, x^{(h)}, y) = -\sum_{t=1}^{T} \log p(x_t^{(h)} \mid x_{<t}^{(h)}, x^{(p)}, y) \tag{1}$$

At inference time, the prediction is made as follows:

$$\underset{y \in Y}{\operatorname{argmax}} \log p(y) + \sum_{t=1}^{T} \log p(x_t^{(h)} \mid x_{<t}^{(h)}, x^{(p)}, y) \tag{2}$$

Throughout all of the experiments in this paper, we assume a uniform label prior $p(y)$, so $p(y)$ will not affect the argmax in Eq. (2) and can be omitted.

3.1 Parameterization

Our model, which we refer to as GenNLI, is parameterized with a standard RNN-based sequence-to-sequence architecture with attention and a copy mechanism between the encoder and the decoder.[1]

Encoder. Our encoder uses a standard bidirectional recurrent neural network (RNN) using long short-term memory (LSTM; Hochreiter and Schmidhuber, 1997):

$$\mathbf{s}_n = [f_{e_1}(\mathbf{v}_n, \overrightarrow{\mathbf{s}_{n-1}}); f_{e_2}(\mathbf{v}_n, \overleftarrow{\mathbf{s}_{n+1}})]$$

where f_{e_1} and f_{e_2} are forward and backward LSTM recurrences, respectively, $\mathbf{v}_n$ is the word embedding of $x_n^{(p)}$, and $\mathbf{s}_n$ is the concatenation of the forward and backward RNN hidden states at position n in the premise.

Decoder. Our decoder uses an RNN with dot product attention from Luong et al. (2015) and a copy mechanism (Gu et al., 2016). The decoder hidden state at step t is computed as $\mathbf{h}_t = f_d(\mathbf{w}_t, \mathbf{h}_{t-1})$.

[1]We also experimented with transformer architectures (Vaswani et al., 2017) and found similar results.

where f_d is the forward LSTM recurrence in the decoder and $\mathbf{w}_t$ is the word embedding of $x_t^{(h)}$. The word distribution at position $t+1$ is computed as follows:

$$\mathbf{p}_{vocab} = \operatorname{softmax}(\mathbf{V}'(\mathbf{V}[\mathbf{h}_t, \mathbf{s}_t^*, \mathbf{v}_y] + b) + b')$$

where $\mathbf{v}_y$ is the label embedding of y, $\mathbf{s}_t^*$ is the context vector at step t computed using attention (full details of the attention mechanism are omitted for brevity but can be found in Luong et al., 2015), and $\mathbf{V}$, $\mathbf{V}'$, b, and b' are learnable parameters. Note the presence of the label embedding $\mathbf{v}_y$ concatenated to $\mathbf{h}_t$ and $\mathbf{s}_t^*$ to form the input to the $\operatorname{softmax}$ layer. This enables the label to directly influence the word distribution. We also use label-specific beginning-of-sentence (BOS) tokens as the initial symbol fed to the decoder RNN. Concretely, we create the embeddings for all BOS symbols BOS_y ($y \in Y$) and prepend $BOS_{y'}$ to the hypothesis where y' is the label for the instance.

Copy mechanism. In some datasets, hypotheses are written by humans when provided a premise and label (Bowman et al., 2015). We observed that these hypotheses sometimes appear to be written by slightly modifying the premise according to the label, e.g., adding "not" to negate the premise, or by replacing a phrase with a phrasal hypernym, such as replacing "soccer game" with "sport" (Marelli et al., 2014; Bowman et al., 2015). The tokens in a premise/hypothesis pair often show a large degree of overlap. So we use a copy mechanism (Gu et al., 2016) to (1) reduce the difficulty of word prediction when training sequence-to-sequence models on small datasets and (2) encourage the model to pay more attention to the token differences between the textual input of the encoder and decoder. We compute:

$$p_{copy} = \sigma(\mathbf{w}_{copy}^{\top}[\mathbf{h}_t, \mathbf{s}_t^*, \mathbf{v}_y] + b_{copy}) \tag{3}$$

where $p_{copy} \in [0, 1]$ is the probability of copying a word from the input sequence, the vector $\mathbf{w}_{copy}$ and scalar b_{copy} are learnable parameters, and σ represents the logistic sigmoid function. We use an extended vocabulary for a specific sentence pair which includes all the words appearing in the input sentence so that the decoder can copy specific words from the input sentence instead of generating out-of-vocabulary (OOV) words.

$$
\begin{aligned}
\text{perceptron loss:}\quad & -\log p(x^{(h)} \mid x^{(p)}, y) + \max_{y' \in Y} \log p(x^{(h)} \mid x^{(p)}, y') \\
\text{hinge loss:}\quad & -\log p(x^{(h)} \mid x^{(p)}, y) + \max_{y' \in Y} \{\log p(x^{(h)} \mid x^{(p)}, y') + \text{cost}(y, y')\} \\
\text{log loss:}\quad & -\log p(x^{(h)} \mid x^{(p)}, y) + \log \sum_{y' \in Y} p(x^{(h)} \mid x^{(p)}, y') \\
\text{softmax-margin:}\quad & -\log p(x^{(h)} \mid x^{(p)}, y) + \log \sum_{y' \in Y} \exp\{\log p(x^{(h)} \mid x^{(p)}, y') + \text{cost}(y, y')\} \\
\text{Bayes risk:}\quad & \mathbb{E}_{p(y' \mid x^{(h)}, x^{(p)})}[\text{cost}(y, y')] = \sum_{y' \in Y} \text{cost}(y, y') \frac{p(x^{(h)} \mid x^{(p)}, y')}{\sum_{y'' \in Y} p(x^{(h)} \mid x^{(p)}, y'')} \\
\text{infinilog loss:}\quad & -\log p(x^{(h)} \mid x^{(p)}, y) + \log \sum_{y' \in Y, y' \neq y} p(x^{(h)} \mid x^{(p)}, y')
\end{aligned}
$$

Table 1: Discriminative objectives considered for fine-tuning GenNLI in this paper. Each is defined for a single training example $\langle x^{(p)}, x^{(h)}, y \rangle$, where $x^{(p)}$ is the premise, $x^{(h)}$ is the hypothesis, and $y \in Y$ is the label.

4 Discriminative Fine-Tuning

Lewis and Fan (2019) showed that generative classifiers for question answering can be improved by a discriminative fine-tuning step after estimating the generative classifier distributions. They used log loss as their discriminative objective. We also consider using a discriminative fine-tuning step when training our model, specifically we compare log loss to four other discriminative losses:

- **Perceptron loss**: the loss function underlying the perceptron algorithm (Rosenblatt, 1958)
- **Hinge loss**: the loss function underlying support vector machines (SVMs) and structured SVMs (Wahba et al., 1999; Taskar et al., 2004)
- **Softmax-margin**: which combines log loss with a cost function as in hinge loss (Povey et al., 2008; Gimpel and Smith, 2010)
- **Bayes risk**: the expectation of the cost function with respect to the model's conditional distribution (Kaiser et al., 2000; Smith and Eisner, 2006)

Table 1 shows these discriminative losses.[2] Some losses use a **cost function**, which can be chosen by the practitioner to penalize different errors differently. In our experiments, we define it as $\text{cost}(y, y') = 1$ for $y \neq y'$ and $\text{cost}(y, y') = 0$ if $y = y'$, where y is the gold label and y' is a candidate label.

In addition, we introduce a very simple loss that is inspired by these other discriminative losses while performing quite well overall in our experiments. We call it the **infinilog loss** and define it as follows:

$$
-\log p(x^{(h)} \mid x^{(p)}, y) + \log \sum_{\substack{y' \in Y \\ y' \neq y}} p(x^{(h)} \mid x^{(p)}, y') \tag{4}
$$

The infinilog loss is different from log loss in that the gold label is excluded from the sum. Therefore, infinilog is not bounded below by zero, unlike all other discriminative losses we consider. It does not approach zero as the model becomes increasingly confident in the correct classification, as is the case with log loss and softmax-margin. Rather, infinilog is unbounded, causing learning to continually seek to increase the score of the correct label and decrease the score of the incorrect labels.

We can view infinilog as softmax-margin with a cost function that returns $-\infty$ when $y = y'$ and 0 otherwise. However, the convention usually assumed when defining cost functions for softmax-margin is for the cost function to be nonnegative (Gimpel and Smith, 2010), and similar conventions are assumed with hinge loss. So we choose to use a distinct name for this loss.

Our results in Section 7 show that fine-tuning using infinilog or one of the investigated discriminative losses leads to better performance than log loss fine-tuning, which was proposed for generative classifiers by Lewis and Fan (2019).

Though the above objectives appear discriminative due to their direct penalization of incorrect labels, they do so by using the key building blocks of generative classifiers. Thus, this fine-tuning achieves some of the benefits of discriminative classifiers while retaining the advantages of generative classifiers, as shown for question answering by Lewis and Fan (2019) and also shown in our experiments below.

[2] Again, the label prior $p(y)$ ends up canceling out because it is uniform over labels, so we do not show it.

5 Experiments

5.1 Datasets

We experiment with five sentence pair datasets, namely the Stanford Natural Language Inference corpus (SNLI; Bowman et al., 2015), the SICK dataset (Marelli et al., 2014), the Multi-Genre Natural Language Inference corpus (MultiNLI; Williams et al., 2018), the binary Recognizing Textual Entailment (RTE; Dagan et al., 2005) dataset from the GLUE benchmark (Wang et al., 2018), and the Microsoft Research Paraphrase Corpus (MRPC; Dolan et al., 2004) also from GLUE.[3] The statistics of the datasets can be found in the Appendix. For MultiNLI, we use the matched dev set and mismatched dev set as our validation and test sets, respectively. Otherwise, we use the standard train, validation, and test splits from the original papers (for SNLI and SICK) or the GLUE benchmark (for RTE and MRPC).[4]

5.2 Baseline Models

We compare our GenNLI model to two baseline discriminative models, and three pretrained models as described below.

We consider InferSent (Conneau et al., 2017) and ESIM (Chen et al., 2017a) as our discriminative baselines. InferSent uses a BiLSTM network with max pooling (Collobert and Weston, 2008) to learn generic sentence embeddings that perform well on several NLI tasks. ESIM has a relatively complicated network structure, including a recursive architecture of local inference modeling (MacCartney, 2009; Parikh et al., 2016) and inference composition. The pretrained models we compare to are BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and XLNet (Yang et al., 2019).

We select these models as our baselines because (1) they are open-source and are frequently used as baselines for NLI tasks in related work (Peters et al., 2018; Williams et al., 2018), and (2) their performance is strong on standard leaderboards.[5]

[3]While MRPC is a binary paraphrase classification task rather than an NLI or entailment task, we treat it as a binary entailment task by choosing one of the sentences arbitrarily as the premise and using the other as the hypothesis.

[4]MRPC and RTE have no public test set, so we report their performances on the development sets.

[5]GLUE leaderboard: https://gluebenchmark.com/leaderboard/; SNLI leaderboard: https://nlp.stanford.edu/projects/snli/

5.3 Training Details

Both generative and discriminative models are initialized with GloVe pretrained word embeddings (Pennington et al., 2014).[6] The word embedding dimension and the LSTM hidden state dimension are set to 300. All parameters, including the word embeddings, are updated during training. The label embedding dimensionality for GenNLI is set to 100. All the experiments are conducted 5 times with different random seeds and we report the median scores.

GenNLI. The training includes two steps: the model is first trained with the generative objective only (Equation 1) for 20 epochs, followed by the discriminative fine-tuning objective only (one of the objectives in Table 1) for 15 epochs. Unless otherwise specified, we use infinilog for discriminative fine-tuning. Section 7 compares fine-tuning objectives.[7]

Discriminative baselines. We run the open source code of InferSent[8] and ESIM.[9] Following their implementation, training stops when the performance on the dev set does not improve across 5 consecutive epochs or the learning rate sufficiently decays (e.g,. less than e^{-5}).

For both GenNLI and discriminative baselines, we use the Adam (Kingma and Ba, 2015) optimizer with learning rates of 0.001 and 0.1, and SGD with learning rates 0.1, 0.5, 1, and 2, and select the model with the best performance on the dev set.

Pretrained baselines. We use the Hugging Face PyTorch implementation (Wolf et al., 2019) of pretrained transformer (Vaswani et al., 2017) models.[10] BERT, XLNet, and RoBERTa are configured with 'bert-base-uncased', 'xlnet-base-cased', and 'roberta-base', respectively. We use the vector at the position of the [CLS] token in the last layer as the output of pretrained models, and map the output to NLI classification with a linear transformation. We fine-tune the pretrained models on our training sets for 10 epochs. We observe that the models usually converge within the first 3-5 epochs.

[6]All of our experiments use uncased 300-dimensional GloVe vectors trained on 6 billion tokens (http://nlp.stanford.edu/data/glove.6B.zip).

[7]Our implementation is available at https://github.com/tyliupku/gen-nli

[8]github.com/facebookresearch/InferSent

[9]github.com/coetaur0/ESIM

[10]github.com/huggingface/transformers

	5	20	100	500	1000	all
			SNLI			
GenNLI	**43.5**	**45.6**	**50.6**	60.6	64.2	82.2
InferSent	37.5	39.6	44.1	56.0	63.9	84.5
ESIM	38.4	38.6	46.7	58.2	65.4	87.6
BERT	33.4	37.3	47.4	70.1	78.7	90.6
XLNet	34.1	35.6	45.1	72.3	77.3	90.9
RoBERTa	35.1	36.0	49.3	**75.9**	**82.8**	**91.7**
			MNLI			
GenNLI	**44.1**	**47.1**	**49.0**	60.6	63.4	67.5
InferSent	34.1	33.7	35.2	44.9	47.9	70.4
ESIM	36.9	35.4	40.5	49.8	54.2	76.7
BERT	33.0	34.9	41.6	63.6	68.5	83.3
XLNet	35.6	35.6	39.7	68.2	74.4	86.3
RoBERTa	33.2	34.9	42.7	**68.8**	**74.6**	**87.3**
			SICK			
GenNLI	**50.6**	**64.7**	**68.7**	75.2	-	80.4
InferSent	35.5	46.3	60.2	73.2	-	83.6
ESIM	34.5	48.4	62.9	75.4	-	84.6
BERT	36.7	56.7	63.6	78.6	-	86.0
XLNet	34.1	55.3	62.3	79.0	-	86.8
RoBERTa	33.5	56.7	66.3	**83.4**	-	**88.5**
			RTE			
GenNLI	**57.0**	**57.7**	**59.2**	**60.4**	61.4	62.6
InferSent	49.5	47.3	52.4	54.2	55.2	56.3
ESIM	50.1	50.3	53.5	55.8	57.3	58.9
BERT	47.3	48.0	49.1	59.9	64.3	66.4
XLNet	50.9	53.4	55.9	60.3	64.6	68.6
RoBERTa	52.7	53.1	53.8	59.6	**67.8**	**74.7**
			MRPC			
GenNLI	**62.8**	64.1	66.2	67.8	69.9	72.9
InferSent	52.5	54.6	58.1	65.1	70.9	73.1
ESIM	54.1	54.3	59.7	64.8	71.2	75.1
BERT	53.1	55.0	57.0	69.6	74.1	82.3
XLNet	55.3	64.7	**68.5**	78.7	82.5	85.2
RoBERTa	59.8	**65.3**	67.5	**80.3**	**84.4**	**87.1**

Table 2: Comparison of classification accuracy of GenNLI, discriminative baselines, and pretrained baselines with various amounts of training data. Here 5/20/100/500/1000 indicates the number of training instances per class. The best result for each task and data amount is shown in bold, and the best result between GenNLI and the discriminative baselines is underlined.

Accuracy		50%	30%	10%	0%
MRPC	InferSent	40.6	61.7	72.2	73.1
	RoBERTa	66.5	**76.8**	**85.3**	**87.1**
	GenNLI	**68.5**	70.0	71.7	72.9
RTE	InferSent	50.4	50.9	54.5	56.3
	RoBERTa	52.0	**63.5**	**76.2**	**74.7**
	GenNLI	**58.8**	59.9	59.6	62.6
MCC		50%	30%	10%	0%
MRPC	InferSent	-0.018	0.189	0.357	0.379
	RoBERTa	0.000	**0.447**	**0.664**	**0.707**
	GenNLI	**0.214**	0.245	0.303	0.352
RTE	InferSent	0.024	0.111	0.017	0.129
	RoBERTa	0.030	**0.266**	**0.521**	**0.501**
	GenNLI	**0.173**	0.190	0.191	0.230

Table 3: Classification accuracy and Matthews Correlation Coefficient (MCC) when using noisy training sets. The percentages are the fractions of training instances with flipped labels. 0% is the unchanged training set. The best result for each task and each noisy setting is shown in bold, and the second-best one is underlined.

6 Results

6.1 Data Efficiency

We first empirically characterize GenNLI, discriminative baselines, and pretrained baselines in terms of data efficiency. We construct smaller training sets by randomly selecting 5, 20, 100, 500, and 1000 instances per class, and then train separate models across these different-sized training sets. Table 2 shows the results.[11]

When using training sets with 100 or fewer instances per class, GenNLI outperforms the pretrained baselines on all datasets except for MRPC. We would hope that pretrained models like BERT would produce generalized text representations that would perform well after fine-tuning with a relatively small number of examples, but here we observe that a thousand or more examples is required to outperform GenNLI on most datasets.

With small training sets, GenNLI also has better performance than the other discriminative baselines, though the performance gap does shrink as the training set gets larger. The accuracies become comparable when we have 1000 instances per label. We also see that on the full training set, the discriminative baselines outperform GenNLI, which accords with our expectations and the findings of prior work (Ding and Gimpel, 2019).

6.2 Training Label Noise

To measure robustness to label noise, we construct noisy datasets by randomly flipping the labels of 10%, 30%, or 50% of the training instances in the binary classification tasks. The labels of other instances are unchanged. Evaluation is done on the original validation and test sets.

[11] SICK does not have results in the 1000 column because the 'contradiction' label has only 665 instances.

Table 3 shows a comparison of GenNLI, InferSent, and RoBERTa on noisy datasets. In addition, we report the value of the Matthews Correlation Coefficient (MCC) (Matthews, 1975). The value of MCC ranges from -1 to 1, with higher value indicating a better classification model. MCC considers all values in the confusion matrix and describes it with a single number. It is viewed as a balanced measurement when the classes are of very different sizes (Boughorbel et al., 2017).

We find all of the models are robust to slight noise, as the accuracy does not drop dramatically with 10% noisy training data. However, as we increase the proportion of the label noise, the performance of InferSent decreases more rapidly than GenNLI. The results are consistent between the two metrics. It is worth noting that GenNLI works better than RoBERTa under the 50%-noisy-data setting, even though RoBERTa has much stronger performance with the unchanged training set. In other words, GenNLI is more robust as the performance drops only slightly with extremely noisy training data.

In general, training deep neural networks requires abundant clean data. When dealing with potentially noisy data, it may be worthwhile to build both generative and discriminative classifiers.

6.3 Imbalanced Label Distributions

We also perform experiments in a setting with label imbalance in the training set. Each imbalanced training set is constructed by random sampling and keeping only 10%, 20%, or 50% of the instances from one selected class, and keeping all the instances from the other classes. We use the original validation and test sets. We still use a uniform prior for GenNLI.

Table 4 shows the comparison of generative, discriminative, and BERT-based classifiers under various imbalanced training sets.[12] Aside from the 10%-non-entailment RTE dataset, RoBERTa always performs the best. This is unsurprising because, even after subsampling, the training set sizes are on a similar order of magnitude as the full sets, with which RoBERTa excels (Table 2). However, RoBERTa does show degradation as the subsampling rate becomes more extreme (more than 10% in MRPC, 8-18% in RTE, and 4-5% on MNLI). GenNLI shows a smaller or comparable decrease in performance, though its overall accuracies are lower. In comparing the generative and discriminative classifiers, GenNLI always outperforms InferSent when keeping only 10% of the instances for the selected class. However, as the percentage of instances in the selected class increases, InferSent begins to perform better than GenNLI.

Another finding is that the different labels have different effects under the imbalanced setting. For example, the performance of RTE/non-entailment decreases more slowly than RTE/entailment for both GenNLI and InferSent, which might suggest that the non-entailment label requires fewer training examples than entailment.

Data efficiency might also affect performance under the label imbalanced setting. We believe it is not the only factor for a performance difference between the generative and discriminative models, as the MNLI dataset has 130k instances per class and the training set still has more than 270k instances in total even under the 10% setting, indicating GenNLI has certain advantages over InferSent when the label distribution is imbalanced.

7 Analysis

7.1 Modeling and Training Decisions

We now empirically assess the importance of major components of modeling and training. As shown in Table 5, the copy mechanism is essential, which meets our expectation because we observe a lot of lexical overlap between the premise and hypothesis in many pairs.[13] We find both generative training and fine-tuning objectives to be helpful, as better results are achieved by training with both objectives.

GenNLI defines the conditional distribution of hypotheses given a premise and label. We could instead model $p(x^{(p)} \mid x^{(h)}, y)$. The final two rows of Table 5 compare the two, showing better performance with $p(x^{(h)} \mid x^{(p)}, y)$. The difference is larger in SNLI, which may be due in part to how the dataset was created. If annotators are provided with a premise and label and asked to write hypotheses, as in SNLI, we would expect that a generative model that matches this process would excel. The difference may also be due to the fact

[12] We report the results on these three datasets since they represent different characteristics in terms of training set size, number of candidate labels, and performance difference between GenNLI and InferSent on the full training set.

[13] All the experiments in our paper are in-domain testing. We also test GenNLI in out-of-domain (OOD) datasets to see whether the copy mechanism is helpful in this case. For example, we train on MNLI and test on SICK. The trend is not consistent across different OOD settings.

Dataset	Subsampled Label	Model	Accuracy				Matthews Correlation Coefficient			
			10%	20%	50%	100%	10%	20%	50%	100%
MRPC	paraphrase	InferSent	49.2	63.1	70.6	73.1	0.244	0.362	0.372	0.379
		RoBERTa	**74.1**	**83.2**	**86.0**	**87.1**	**0.526**	**0.645**	**0.688**	**0.707**
		GenNLI	70.2	70.7	72.0	72.9	0.301	0.367	0.318	0.352
	non-paraphrase	InferSent	68.3	70.9	73.8	73.1	0.191	0.287	0.373	0.379
		RoBERTa	**77.2**	**81.2**	**86.3**	**87.1**	**0.469**	**0.568**	**0.697**	**0.707**
		GenNLI	70.8	70.3	72.2	72.9	0.333	0.292	0.319	0.352
RTE	entailment	InferSent	47.3	47.3	52.3	56.3	0.000	0.036	0.135	0.129
		RoBERTa	**66.7**	**66.7**	**71.5**	**74.7**	**0.226**	**0.230**	**0.426**	**0.501**
		GenNLI	55.8	56.5	59.9	62.6	0.128	0.135	0.194	0.230
	non-entailment	InferSent	52.7	52.7	54.0	56.3	0.001	0.035	0.065	0.129
		RoBERTa	56.0	**62.1**	**72.9**	**74.7**	0.177	**0.371**	**0.471**	**0.501**
		GenNLI	**60.5**	60.3	62.2	62.6	**0.209**	0.204	0.181	0.230
MNLI	entailment	InferSent	57.4	60.1	67.8	70.4	0.396	0.431	0.522	0.557
		RoBERTa	**82.4**	**84.8**	**87.0**	**87.3**	**0.747**	**0.776**	**0.806**	**0.809**
		GenNLI	60.8	61.7	67.1	67.5	0.410	0.452	0.497	0.512
	neutral	InferSent	60.5	62.5	68.8	70.4	0.445	0.469	0.539	0.557
		RoBERTa	**83.0**	**84.5**	**85.9**	**87.3**	**0.754**	**0.769**	**0.790**	**0.809**
		GenNLI	61.7	63.8	67.6	67.5	0.463	0.487	0.491	0.512
	contradiction	InferSent	60.8	64.0	67.9	70.4	0.444	0.479	0.526	0.557
		RoBERTa	**82.7**	**84.5**	**86.6**	**87.3**	**0.748**	**0.773**	**0.800**	**0.809**
		GenNLI	61.0	62.0	65.6	67.5	0.444	0.466	0.492	0.512

Table 4: Classification accuracies and Matthews Correlation Coefficients of test sets when training on label-imbalanced training sets. Column headers indicate the percentage of the subsampled label's training instances that are retained in the training set. All training instances are used for the other labels. The best result for each task and each subsample setting is shown in bold, and the second-best one is underlined.

	SNLI	RTE
GenNLI	82.2	62.6
no copy mechanism	74.4	54.7
no generative training	80.1	60.3
no discriminative fine-tuning	79.1	61.7
GenNLI, $p(x^{(h)} \mid x^{(p)}, y)$	82.2	62.6
GenNLI, $p(x^{(p)} \mid x^{(h)}, y)$	77.1	59.7

Table 5: Results showing contribution of individual modeling/training decisions on SNLI and RTE.

	SNLI			RTE		
	100	1000	all	100	1000	all
perceptron	49.6	62.5	80.4	57.9	60.1	61.1
hinge	49.9	63.1	81.1	58.8	61.3	62.2
log	49.1	62.3	80.7	57.4	59.7	60.5
softmax-margin	**50.6**	**64.2**	81.9	**59.2**	61.1	62.2
infinilog	50.0	63.7	**82.2**	58.1	**61.4**	**62.6**
Bayes risk	49.0	62.6	80.1	58.3	60.6	61.4

Table 6: Comparision of discriminative fine-tuning objectives on SNLI and RTE datasets. The best result for each task and data amount is shown in bold, and the second-best one is underlined.

that in the entailment pairs, the premise often has more information than the hypothesis, and it is expected to be easier to remove information (when generating the hypothesis from the premise) than to add it.

7.2 Discriminative Fine-Tuning Comparison

Table 6 compares discriminative fine-tuning objectives.[14] Several choices, including hinge, softmax-margin, and infinilog, consistently outperform the log loss used as discriminative fine-tuning objective by Lewis and Fan (2019). The perceptron loss and Bayes risk also often outperform log loss. It is worth noting that infinilog performs the best when using the full training set on four out of five datasets (see Appendix for full results), while softmax-margin is best with smaller training sets. These results suggest that improving discriminative fine-tuning does not harm the data efficiency benefits of generative classifiers, but rather is able to accentuate them.

7.3 Data Generation

One advantage of generative models is that they can be used to generate samples in order to inter-

[14] Note that all models are trained with the generative objective before discriminative fine-tuning. Results for other datasets are provided in the Appendix.

GenNLI trained on full SICK training set		
N	$x^{(p)}$	A man is sitting near a bike and is writing a note.
	$x^{(h)}$	A man with paint covered clothes is sitting outside in a busy area writing something.
	gen.	**A man is sitting in a bike and is writing a note in a busy area.**
E	$x^{(p)}$	People wearing costumes are gathering in a forest and are looking in the same direction.
	$x^{(h)}$	Masked people are looking in the same direction in a forest.
	gen.	**People wearing costumes are looking in a forest.**
C	$x^{(p)}$	There is no child holding a water gun or getting sprayed with water.
	$x^{(h)}$	A laughing child is holding a water gun and getting sprayed with water.
	gen.	**A child is holding a water gun.**
GenNLI trained on small SICK training set		
N	$x^{(p)}$	A little girl and a woman wearing a yellow shirt are getting splashed by a city fountain.
	$x^{(h)}$	The young girl is playing on the edge of a fountain and an older woman is watching her.
	gen.	**A little girl is playing in the background.**
E	$x^{(p)}$	A man is playing a flute.
	$x^{(h)}$	A man is playing the flute.
	gen.	**A flute is being played by a man.**
C	$x^{(p)}$	There is no man on a rock high above some trees standing in a strange position.
	$x^{(h)}$	A man is on a rock high above some trees and is standing in a strange position.
	gen.	**A man is on a rock high above some trees is standing in a strange position.**

Table 7: Generated hypotheses for premises with given labels (N = neutral, E = entailment, C = contradiction).

pret how the model works. Since we include label information in the decoder of GenNLI, we are able to generate various hypotheses for a premise by specifying the label. Table 7 shows example generations from two models, one using the full dataset for training and the other using a small training set with only 500 examples per class. We use greedy decoding for these generations.

We observe that the generated examples comport with the labels and premises we have specified, and the generation is of high quality in terms of fluency. However, the diversity is relatively low, with the generated samples looking similar to the premise. This is not surprising since we assume the decoder relies heavily on the copy mechanism when trained on NLI pairs, as some hypotheses differ only slightly from their corresponding premises. The generations are relatively short compared to the gold hypotheses, which is likely due in part to greedy decoding. The model might require more training data and/or a different decoding algorithm to be able to produce more diverse generations. We also note that generations for the entailment label generally look better than those for contradiction.[15]

8 Conclusions and Future Work

We proposed GenNLI, a discriminatively-finetuned generative classifier for NLI tasks, and empirically characterized its performance by comparing it to discriminative models and pretrained models. We found several discriminative fine-tuning objectives to outperform log loss, including infinilog, a simple but effective choice. We conducted extensive experiments with GenNLI, showing its robustness across challenging empirical conditions. We also showed its ability to generate hypotheses given premises and particular labels. Future work may explore generating of diverse sets of hypotheses for a given premise and label, with the goal of performing data augmentation. Other future work will be to measure the performance of GenNLI on adversarial and similarly challenging NLI datasets.

Acknowledgments

We would like to thank Sam Wiseman for contributions to an earlier version of this manuscript, and the anonymous reviewers for their helpful feedback. Z. Sui and B. Chang thank NSFC (No. 61876004, No. U19A2065) and Beijing Academy of Artificial Intelligence (BAAI) for their generous support.

References

Sabri Boughorbel, Fethi Jarray, and Mohammed El-Anbari. 2017. Optimal classifier for imbalanced data using matthews correlation coefficient metric. *PloS one*, 12(6):e0177678.

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 (EMNLP)*. Association for Computational Linguistics.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017a. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association*

[15] Future work may consider using these generations for data augmentation. While our preliminary experiments in this direction were not positive, future work will consider fine-tuning pretrained language models as generative classifiers and using them with diverse decoding strategies to automatically expand small training sets.

for Computational Linguistics (Volume 1: Long Papers), pages 1657–1668, Vancouver, Canada. Association for Computational Linguistics.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017b. Recurrent neural network-based sentence encoder with gated attention for natural language inference. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*, pages 36–40, Copenhagen, Denmark. Association for Computational Linguistics.

Jihun Choi, Kang Min Yoo, and Sang-goo Lee. 2018. Learning to compose task-specific tree structures. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

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.

I. Dagan, D. Roth, F. Zanzotto, and M. Sammons. 2013. *Recognizing Textual Entailment: Models and Applications*. Morgan & Claypool.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Xiaoan Ding and Kevin Gimpel. 2019. Latent-variable generative models for data-efficient text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 507–517, Hong Kong, China. Association for Computational Linguistics.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, pages 350–356, Geneva, Switzerland.

Kevin Gimpel and Noah A. Smith. 2010. Softmax-margin CRFs: Training log-linear models with cost functions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 733–736, Los Angeles, California. Association for Computational Linguistics.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. Association for Computational Linguistics.

Yichen Gong, Heng Luo, and Jian Zhang. 2018. Natural language inference over interaction space. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *NAACL*, pages 107–112.

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

J. Kaiser, B. Horvat, and Z. Kacic. 2000. A novel loss function for the overall risk criterion based discriminative training of HMM models. In *Proc. of ICSLP*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Mike Lewis and Angela Fan. 2019. Generative question answering: Learning to answer the whole question. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Tianyu Liu, Xin Zheng, Baobao Chang, and Zhifang Sui. 2020a. HypoNLI: Exploring the artificial patterns of hypothesis-only bias in natural language inference. *arXiv preprint arXiv:2003.02756*.

Tianyu Liu, Xin Zheng, Xiaoan Ding, Baobao Chang, and Zhifang Sui. 2020b. An empirical study on model-agnostic debiasing strategies for robust natural language inference. In *Proceedings of the 24th Conference on Computational Natural Language Learning (CoNLL)*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Bill MacCartney. 2009. Natural language inference. In *P.hD Thesis*.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 216–223, Reykjavik, Iceland. European Languages Resources Association (ELRA).

Brian W Matthews. 1975. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451.

R Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *arXiv preprint arXiv:1902.01007*.

Pasquale Minervini and Sebastian Riedel. 2018. Adversarially regularising neural NLI models to integrate logical background knowledge. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 65–74, Brussels, Belgium. Association for Computational Linguistics.

Andrew Y. Ng and Michael I. Jordan. 2002. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In *Advances in neural information processing systems*, pages 841–848.

Yixin Nie and Mohit Bansal. 2017. Shortcut-stacked sentence encoders for multi-domain inference. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*, pages 41–45, Copenhagen, Denmark. Association for Computational Linguistics.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, Austin, Texas. Association for Computational Linguistics.

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 Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In **SEM@NAACL-HLT*, pages 180–191.

D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah. 2008. Boosted MMI for model and feature space discriminative training. In *Proc. of ICASSP*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Frank Rosenblatt. 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

David A. Smith and Jason Eisner. 2006. Minimum risk annealing for training log-linear models. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 787–794, Sydney, Australia. Association for Computational Linguistics.

Ben Taskar, Carlos Guestrin, and Daphne Koller. 2004. Max-margin markov networks. In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 25–32. MIT Press.

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 I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Grace Wahba et al. 1999. Support vector machines, reproducing kernel hilbert spaces and the randomized gacv. *Advances in Kernel Methods-Support Vector Learning*, 6:69–87.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Wei Wu, Houfeng Wang, Tianyu Liu, and Shuming Ma. 2018. Phrase-level self-attention networks for universal sentence encoding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3729–3738.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 5753–5763. Curran Associates, Inc.

Dani Yogatama, Chris Dyer, Wang Ling, and Phil Blunsom. 2017. Generative and discriminative text classification with recurrent neural networks. *arXiv preprint arXiv:1703.01898*.

A Appendix

A.1 Dataset

We present our results on the five publicly available NLI datasets shown in Table 8, which include the Stanford Natural Language Inference (SNLI) corpus (Bowman et al., 2015), the SICK corpus (Marelli et al., 2014), the Multi-Genre Natural Language Inference corpus (MultiNLI) (Williams et al., 2018), the Recognizing Textual Entailment (RTE) (Dagan et al., 2005) corpus, and the Microsoft Research Paraphrase Corpus (MRPC) from the GLUE benchmark (Wang et al., 2018).[16] For MultiNLI, we use the matched dev set and mismatched dev set as our validation and test sets, respectively. Table 8 shows the statistics of the datasets in our paper.

[16]For the corpora with no public test set, we report the performance on the dev set in our paper.

Dataset	#Train	#Valid	#Test	#Class
SNLI	549K	9.8K	9.8K	3
MultiNLI	392K	9.8K	9.8K	3
SICK	4.5K	0.5K	4.9K	3
RTE	2.4K	0.2K	-	2
MRPC	4.0K	1.7K	-	2

Table 8: Dataset statistics.

We use the standard train, validation, and test divisions from the original papers (SNLI, MultiNLI and SICK) or GLUE benchmark (RTE and MRPC). These datasets can be downloaded at `https://nlp.stanford.edu/projects/snli/`, `https://gluebenchmark.com`, and `http://marcobaroni.org/composes/sick.html`.

A.2 Discriminative Fine-Tuning Comparison

Table 9 lists the full comparison results of different discriminative fine-tuning objectives. Several choices, including hinge, softmax-margin, and infinilog, consistently outperform the log loss used as discriminative fine-tuning objective by Lewis and Fan (2019). It is worth noting that infinilog performs the best when using the full training set on four out of five datasets.

A.3 Data Generation

Table 10 shows example generations from two models, one using the full dataset for training and the other using a small training set with only 500 examples per class.

A.4 Ablation of Copy Mechanism in Generation

Table 11 shows the generated hypotheses of the proposed generative classifier. Comparing the generative classifiers with and without copy mechanism, we find that the copy mechanism can help the model capture key differences between premise and hypothesis sentences given the specified labels. For example, we see 'There is no child' versus 'A child' given the label 'contradiction', and 'another animal' versus 'a brown dog' given the label 'neutral'. The copy mechanism also helps to avoid excessive semantic drift, e.g., generating the same subject as the premise and maintaining a reasonable amount of text with the premise.

Although classification accuracy increases by adopting discriminative finetuning after generative training, the finetuning method can lead to ungrammatical or repetitive generated sentences, as demon-

	5	20	100	500	1000	all
SNLI						
perceptron	41.8	44.1	49.6	58.4	62.5	80.4
hinge	42.3	45.3	49.9	58.6	63.1	81.1
log	42.1	43.2	49.1	58.6	62.3	80.7
softmax-margin	**43.5**	45.3	**50.6**	**60.6**	**64.2**	81.9
infinilog	42.7	**45.6**	50.0	59.8	63.7	**82.2**
Bayes risk	42.8	44.7	49.0	58.3	62.6	80.1
MNLI						
perceptron	42.7	45.5	46.7	58.1	61.6	66.3
hinge	43.2	46.3	48.2	60.2	62.8	67.1
log	42.1	45.4	46.7	58.3	61.4	66.2
softmax-margin	**44.1**	**47.1**	**49.0**	**60.6**	**63.4**	**67.5**
infinilog	42.3	45.9	47.7	60.0	62.8	67.3
Bayes risk	43.1	45.7	47.7	59.1	61.6	66.2
SICK						
perceptron	49.1	61.7	66.9	73.4	-	79.7
hinge	**50.6**	63.8	67.8	73.6	-	80.0
log	48.6	62.1	67.5	73.1	-	79.8
softmax-margin	50.2	**64.7**	**68.7**	74.3	-	80.2
infinilog	48.4	62.4	68.3	**75.2**	-	**80.4**
Bayes risk	48.2	62.4	67.2	72.8	-	79.7
RTE						
perceptron	56.1	57.4	57.9	59.4	60.1	61.1
hinge	56.4	57.1	58.8	59.2	61.3	62.2
log	56.5	57.1	57.4	59.1	59.7	60.5
softmax-margin	**57.0**	**57.7**	**59.2**	**60.4**	61.1	62.2
infinilog	56.7	57.4	58.1	59.6	**61.4**	**62.6**
Bayes risk	56.1	57.2	58.3	59.3	60.6	61.4
MRPC						
perceptron	62.1	62.5	64.6	66.1	68.6	69.8
hinge	62.3	63.8	65.4	67.1	69.0	71.8
log	61.7	62.1	64.1	65.9	68.1	71.3
softmax-margin	62.6	**64.1**	**66.2**	**67.8**	**69.9**	72.8
infinilog	62.8	63.7	65.6	67.4	69.8	**72.9**
Bayes risk	**63.2**	63.5	65.6	67.7	69.5	72.5

Table 9: Comparison of discriminative fine-tuning objectives. The best result for each task and data amount is shown in bold, and the second-best one is underlined.

strated in Table 11. This shows that generated text with higher quality does not necessarily lead to better performance in NLI classification.

		GenNLI trained on full RTE training set
E	$x^{(p)}$	Only a week after it had no comment on upping the storage capacity of its hotmail e-mail service , microsoft early thursday announced it was boosting the allowance to 250mb to follow similar moves by rivals such as google , yahoo , and lycos.
	$x^{(h)}$	Microsoft 's hotmail has raised its storage capacity to 250mb.
	gen.	**Microsoft was boosting of its hotmail e-mail.**
N	$x^{(p)}$	The name for the newest james bond film has been announced today . the 22nd film , previously known only as " bond 22 " , will be called " quantum of solace " . Eon productions who are producing the film made the announcement today at pinewood studios , where production for the film has been under way since last year . The name of the film was inspired by a short story of the same name from for your eyes only by bond creator , ian fleming.
	$x^{(h)}$	James bond was created by ian fleming.
	gen.	**James bond is a member of the film.**
		GenNLI trained on small RTE training set
E	$x^{(p)}$	Lin piao , after all , was the creator of mao 's " little red book " of quotations.
	$x^{(h)}$	Lin piao wrote the " little red book " .
	gen.	**Lin piao 's " little red book '.**
N	$x^{(p)}$	A dog is pushing a toddler into a rain puddle.
	$x^{(h)}$	A dog is pulling a toddler out of a rain puddle.
	gen.	**A dog is pushing a rain puddle.**

Table 10: Generated hypotheses for premises with given labels (N = not entailment, E = entailment).

Neutral	$x^{(p)}$	A brown dog is attacking another animal in front of the man in pants.
	$x^{(h)}$	Two dogs are fighting.
	gen.	**A brown dog is attacking a brown dog in front of the man.**
	gen. w/ finetune	**A man is sitting on a black shirt is standing on a black shirt.**
	gen. w/o copy	**A man is wearing a black shirt and is sitting on a dirt ball.**
Entailment	$x^{(p)}$	A group of children in uniforms is standing at a gate and one is kissing the mother.
	$x^{(h)}$	A group of children wearing the same clothes is waiting at a gate and one is kissing the mother
	gen.	**A group of children in uniforms is standing at a gate.**
	gen. w/ finetune	**A group in uniforms at uniforms is gate and one is kissing mother.**
	gen. w/o copy	**A man is sitting on a ball in the water.**
Contradiction	$x^{(p)}$	There is no child holding a water gun or getting sprayed with water.
	$x^{(h)}$	A laughing child is holding a water gun and getting sprayed with water.
	gen.	**A child is holding a water gun.**
	gen. w/ finetune	**There is child child holding a water gun with water.**
	gen. w/o copy	**A dog is jumping in the water.**

Table 11: Generated hypotheses for premises with given labels using models trained on the full SICK dataset. When generating using the discriminatively-finetuned model, the outputs show more repetition, while without the copy mechanism, they drift more from the premise.