Supplementary Material: Latent-Variable Generative Models for Data-Efficient Text Classification

Xiaoan Ding[1] **Kevin Gimpel**[2]

[1]University of Chicago, Chicago, IL, 60637, USA
[2]Toyota Technological Institute at Chicago, Chicago, IL, 60637, USA

xiaoanding@uchicago.edu, kgimpel@ttic.edu

A Comparison of Generative and Discriminative Classifiers

To indicate we have built a strong baseline, we first compare our implementation of the generative and discriminative classifiers to prior work. Note that here we use the whole training set without any truncation on the sequence length.

Table 1 shows that our standard generative and discriminative LSTM models are comparable with Yogatama et al. (2017). All other well-performing models listed in the table are discriminatve models that use more complex modeling methods such as attention to boost performance. Since the focus of our paper is the impact of adding latent variables to generative models, we do not use more complex techniques when building our baselines.

Our results show that our standard generative and discriminative LSTM models are comparable with Yogatama et al. (2017). We also see that the generative models have lower classification accuracies with the full training set, which agrees with the findings in the prior work.

B More Details about Learning with Expectation-Maximization

EM provides a general purpose local search algorithm for learning parameters in probabilistic models with latent variables, and it has been widely used in much prior work and has shown its efficacy in terms of convergence (Ruder et al., 2018; Neal and Hinton, 1998; Dempster et al., 1977).

Salakhutdinov et al. (2003) theoretically study the close relationship between EM and direct optimization approaches with gradient-based methods. Here we empirically characterize the performance of our auxiliary latent generative classifiers with different training methods, namely EM and stochastic gradient descent (SGD) (Bottou, 2010) for direct optimization.

For our latent generative classifiers, the Expectation (E) step computes the posterior distributions over the latent variable:

$$\hat{p}(c \mid x, y) \leftarrow \frac{p(x, y, c)}{\sum_{c' \in \mathcal{C}} p(x, y, c')}$$

The Maximization (M) step seeks to find new parameter estimates by maximizing the following:

$$\sum_{\langle x,y \rangle \in \mathcal{D}} \sum_{c \in \mathcal{C}} \hat{p}(c \mid x, y) \log p(x, y, c)$$

C Alternative Inference Criteria

The classification accuracies of the auxiliary latent generative model in the main text are based on predictions made while marginalizing out the latent variable. In addition, we experiment with two other inference objectives. One uses the posterior $p(c \mid x, y)$ instead of the learned prior $p_\Phi(c)$ during marginalization:

$$\hat{y} = \operatorname*{argmax}_{y \in \mathcal{Y}} \sum_{c \in \mathcal{C}} p_\Theta(x \mid c, y) p(c \mid x, y) p_\Psi(y)$$

The other way is to predict by maximizing the latent variable:

$$\hat{y} = \operatorname*{argmax}_{y \in \mathcal{Y}} \max_{c \in \mathcal{C}} p_\Theta(x \mid c, y) p_\Phi(c) p_\Psi(y)$$

We find very similar performance with all three inference criteria, which agrees with our observation that the classifiers learn peaked prior and posterior distributions over the discrete latent variables.

D Additional Results with Training Sizes

While the main paper contained these results in plots, for completeness we provide the numerical classification accuracies of the discriminative (**Disc.**), generative (**Gen.**), and latent-variable generative (**Lat.**) classifiers trained with various training sizes in Tables 2, 3, 4, 5, 6, and 7.

models	Yelp P	Yelp F	AGNews	Sogou	Yahoo	DBpedia
generative classifier (ours, shared-LSTM)	92.61	57.36	89.88	89.57	68.87	96.46
discriminative classifier (ours)	96.37	65.81	90.09	96.43	73.10	98.78
gen LSTM-shared (Yogatama et al., 2017)	88.20	52.70	90.60	90.30	69.30	95.40
gen LSTM-independent (Yogatama et al., 2017)	90.00	51.90	90.70	93.50	70.50	94.80
disc model (Yogatama et al., 2017)	92.60	59.60	92.10	94.90	73.70	98.70
bag of words (Zhang et al., 2015)	92.20	58.00	88.80	92.90	68.90	96.60
fastText (Joulin et al., 2017)	95.70	63.90	92.50	96.80	72.30	98.60
char-CRNN (Xiao and Cho, 2016)	94.50	61.80	91.40	95.20	71.70	98.60
very deep CNN (Conneau et al., 2017)	95.70	64.70	91.30	96.80	73.40	98.70

Table 1: Summary of the results on the full datasets. Our implementation of the generative model share parameters among classes.

# per class	Disc.	Gen.	Lat.
5	61.97	55.42	62.53
20	64.19	59.06	66.67
100	66.72	69.80	73.50
1k	77.53	78.62	81.22
2k	80.48	80.98	82.96
5k	82.62	83.60	84.97
10k	85.83	85.41	85.65
all	92.20	87.51	87.38

Table 2: Comparison of classification accuracy on Yelp Review Polarity dataset.

# per class	Disc.	Gen.	Lat.
5	23.38	21.67	27.16
20	25.12	26.20	31.78
100	29.82	35.87	38.67
1k	42.85	43.36	46.05
2k	46.09	44.16	48.58
5k	52.23	47.75	49.86
10k	52.23	50.30	50.19
all	59.00	52.34	51.14

Table 3: Comparison of classification accuracy on Yelp Review Full dataset.

# per class	Disc.	Gen.	Lat.
5	40.20	35.12	47.46
20	43.68	37.86	61.45
100	62.58	68.70	78.58
1k	78.08	84.08	86.12
2k	80.80	86.70	87.25
5k	84.87	88.88	89.26
10k	87.25	89.67	89.63
all	89.79	90.00	90.14

Table 4: Comparison of classification accuracy on AG News dataset.

# per class	Disc.	Gen.	Lat.
5	41.75	39.89	61.19
20	52.80	66.32	72.18
100	69.18	77.88	81.48
1k	83.83	84.81	86.40
2k	85.04	86.50	86.61
5k	87.90	87.42	86.62
10k	89.94	87.67	86.81
all	93.40	87.95	86.95

Table 5: Comparison of classification accuracy on Sogou dataset.

E Dataset Description

We present our results on six publicly available text classification datasets introduced by Zhang et al. (2015), which include news categorization, sentiment analysis, question/answer topic classification, and article ontology classification. Dataset names and statistics are shown in Table 8. For each dataset, we randomly pick 5000 instances from the training set as the development set.

F Total Number of Parameters

Table 9 shows the hyperparameter settings for our classifiers. There are various choices of latent variable values and dimensionalities. We select the ones with the best classification accuracy according to the development sets.

Table 10 lists the number of parameters in each classifier. It is related to the discussion about effect of latent structure in the main paper. We created **Gen. PC** and **Lat. PC** to demonstrate that the performance gains are due to the latent-variable structure instead of an increased number of parameters when adding the latent variables.

G Results with Larger Models

We increase the model capacity by increasing dimensionality of the word embedding, LSTM hidden embedding, and label embedding to 200 and refer to the resulting models as the large discriminative (**Disc L.**), generative (**Gen L.**), and latent-variable generative (**Lat L.**) classifiers. Note that we did not change the number of values or dimensionality of the latent variables. We only experimented with two datasets due to GPU memory

# per class	Disc.	Gen.	Lat.
5	13.26	15.39	21.00
20	19.98	30.33	36.55
100	29.97	47.33	50.04
1k	55.15	62.68	64.18
2k	60.83	65.52	65.70
5k	66.07	67.95	67.79
10k	69.00	68.90	67.92
all	72.70	69.14	68.02

Table 6: Comparison of classification accuracy on Yahoo dataset.

# per class	Disc.	Gen.	Lat.
5	32.27	63.02	66.33
20	43.72	82.17	85.56
100	74.73	90.37	92.24
1k	96.11	94.62	95.54
2k	96.85	95.06	95.75
5k	97.76	95.78	96.09
10k	98.15	96.29	96.30
all	98.70	96.73	96.25

Table 7: Comparison of classification accuracy on DBpedia dataset.

limits.[1] Table 11 shows the performance comparison between standard (reported in the main paper) and larger classifiers. We find that the trend **Disc L.** $<$ **Gen L.** $<$ **Lat L.** still holds in most cases in the small-data setting, though the performance gaps shrink as the capacity increases.

References

Léon Bottou. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2017. Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1107–1116, Valencia, Spain. Association for Computational Linguistics.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

[1]The GPU memory consumption is affected by the number of labels in the generative and latent generative classifiers. These two datasets have relatively small numbers of labels.

Dataset	#Train	#Dev	#Test	#Labels
Yelp Polarity	555k	5k	7.6k	2
Yelp Full	645k	5k	50k	5
AGNews	115k	5k	7.6k	4
Sogou	445k	5k	60k	5
Yahoo	1395k	5k	60k	10
DBpedia	555k	5k	70k	14

Table 8: Text classification datasets used in our experiments.

Radford M Neal and Geoffrey E Hinton. 1998. A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer.

Sebastian Ruder, Ryan Cotterell, Yova Kementchedjhieva, and Anders Søgaard. 2018. A discriminative latent-variable model for bilingual lexicon induction. *arXiv preprint arXiv:1808.09334*.

Ruslan Salakhutdinov, Sam Roweis, and Zoubin Ghahramani. 2003. Relationship between gradient and EM steps in latent variable models.

Yijun Xiao and Kyunghyun Cho. 2016. Efficient character-level document classification by combining convolution and recurrent layers. *arXiv preprint arXiv:1602.00367*.

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

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 649–657, Cambridge, MA, USA. MIT Press.

	word embedding	hidden state	label embedding	# latent variable values	latent embedding
Disc.	100	100	100	-	-
Gen.	100	100	100	-	-
Lat.	100	100	100	10, 30, 50	10, 50, 100
Gen. PC	100	100	110	-	-
Lat. PC	100	100	100	10	10

Table 9: Hyperparameter settings of Discriminative (**Disc.**), Generative (**Gen.**), Latent Generative (**Lat.**), Generative PC (**Gen. PC**), Latent Generative PC (**Lat. PC**) classifiers. **PC** stands for "Parameter-comparison Configuration." More description can be found in the main paper.

Dataset	# per class	Disc.	Gen.	Lat.	Gen. PC	Lat. PC
Yelp P	5	4,082,414	12,642,828	16,122,903	12,481,640	12,521,733
	20			14,123,253		
	100			14,123,253		
	1k			14,123,253		
	all			16,122,903		
Yelp F	5	4,082,414	12,642,828	12,522,233	12,481,640	12,521,733
	20			12,522,233		
	100			12,522,233		
	1k			12,522,233		
	all			14,122,553		
AGNews	5	4,082,414	12,642,828	11,430,985	10,104,780	10,137,185
	20			10,137,185		
	100			11,430,985		
	1k			10,137,585		
	all			12,522,333		
Sogou	5	4,082,414	12,642,828	13,162,568	11,634,120	11,671,448
	20			13,163,568		
	100			15,028,468		
	1k			11,676,935		
	all			12,522,233		
Yahoo	5	4,082,414	12,642,828	12,522,533	12,482,520	12,522,533
	20			14,124,053		
	100			12,522,733		
	1k			12,522,533		
	all			16,123,703		
DBpedia	5	4,082,414	12,642,828	12,522,933	12,482,960	12,522,933
	20			14,124,453		
	100			14,124,453		
	1k			16,126,103		
	all			12,522,933		

Table 10: Number of parameters in each classifier.

	# per class	Disc L.	Gen L.	Lat L.	Disc.	Gen.	Lat.
AGNews	5	37.34	40.55	47.91	40.20	35.12	47.46
	20	44.53	47.42	62.36	43.68	37.86	61.45
	100	62.74	76.95	79.63	62.58	68.70	78.58
	1k	80.67	84.82	86.79	78.08	84.08	86.12
	all	90.54	90.16	89.68	89.79	90.00	90.14
Yelp Polarity	5	60.82	60.65	63.07	61.97	55.42	62.53
	20	61.11	64.44	67.50	64.19	59.06	66.67
	100	68.55	71.79	74.37	66.72	69.80	73.50
	1k	77.93	78.91	81.82	77.53	78.62	81.22
	all	92.48	87.76	87.34	92.20	87.51	87.38

Table 11: Comparision of classification accuracies between standard and larger classifiers.