UniDrop: A Simple yet Effective Technique to Improve Transformer without Extra Cost

Zhen Wu[1] Lijun Wu[2] Qi Meng[2] Yingce Xia[2] Shufang Xie[2]
Tao Qin[2] Xinyu Dai[1] Tie-Yan Liu[2]

[1]National Key Laboratory for Novel Software Technology, Nanjing University
[2]Microsoft Research Asia
wuz@smail.nju.edu.cn, daixinyu@nju.edu.cn
{Lijun.Wu,meq,yingce.xia,shufxi,taoqin,tyliu}@microsoft.com

May 12, 2021

Outline

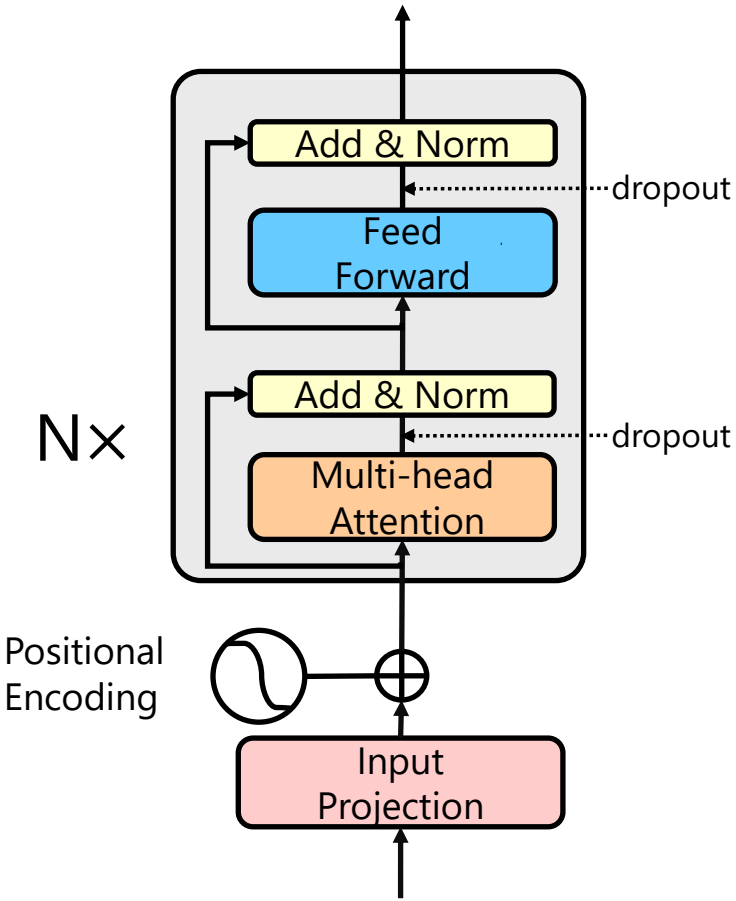
Outline

Background

- Success of Transformer in NLP tasks.
- Auxiliary architectures or external knowledge on Transformer, increasing computational costs or requiring extra resources.
- The over-parameterization of Transformer & Dropout

Background

Transformer Architecture



- Each Transformer block contains:
  - Multi-head Attention sub-layer

$$\texttt{Attn}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \texttt{softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}})\mathbf{V}$$

  - Position-wise Feed-Forward sub-layer

$$\texttt{FFN}(\mathbf{x}) = \max(0, \mathbf{x}\mathbf{W_1} + \mathbf{b_1})\mathbf{W_2} + \mathbf{b_2}$$

  - Each sub-layer is followed by:

$$\texttt{AddNorm}(\texttt{x}) = \texttt{LN}(\texttt{Add}(\texttt{x}))$$

Figure 1: Transformer architecture.

Outline

Motivation

- Pre-experiments show Transformer is still overfitted even equipped with default dropout.

	BLEU
Transformer (default dropout: 0.3)	34.84
+attention dropout, activation dropout	35.46

Table 1: Pre-experiments on IWSLT14 De→En translation task.

Motivation

- Pre-experiments show Transformer is still overfitted even equipped with default dropout.

	BLEU
Transformer (default dropout: 0.3)	34.84
+attention dropout, activation dropout	35.46

Table 1: Pre-experiments on IWSLT14 De→En translation task.

One question:

- Can we achieve stronger or even state-of-the-art (SOTA) results only relying on various dropout techniques instead of extra model architecture design or knowledge enhancement?

Outline

Proposed UniDrop

- `UniDrop` integrates three different-level dropout techniques from fine-grain to coarse-grain, *feature dropout*, *structure dropout*, and *data dropout*, into Transformer models.

  - Feature dropout (FD): conventional dropout (Srivastava et al., 2014), applied on hidden representations of networks.

  - Structure dropout (SD): randomly drops some entire substructures or components from the whole model.

  - Data dropout (DD): randomly drops out some tokens in an input sequence.

Feature Dropout

- **FD-1** (attention dropout): applied to the attention weight $\mathbf{A}$, $\mathbf{A} = \mathbf{Q}\mathbf{K}^\top$.
- **FD-2** (activation dropout): applied after the activation function between the two linear transformations of `FFN` sub-layer.

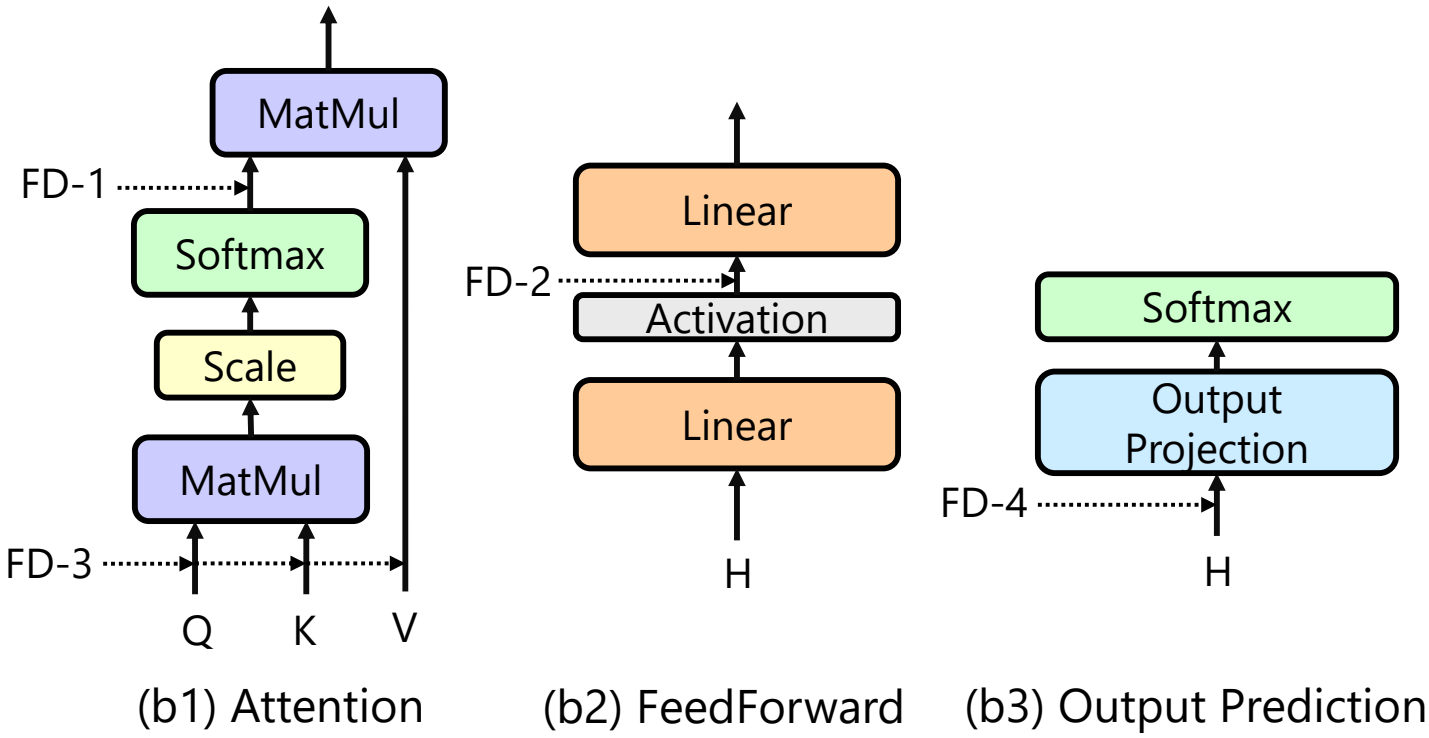


Figure 2: Structure and overview of feature dropout.

Feature Dropout

- **FD-3** (query, key, value dropout): we add dropout to query **Q**, key **K**, and value **V** before calculating attention.
- **FD-4** (output dropout): we also apply dropout to the output features before linear transformation for softmax classification.

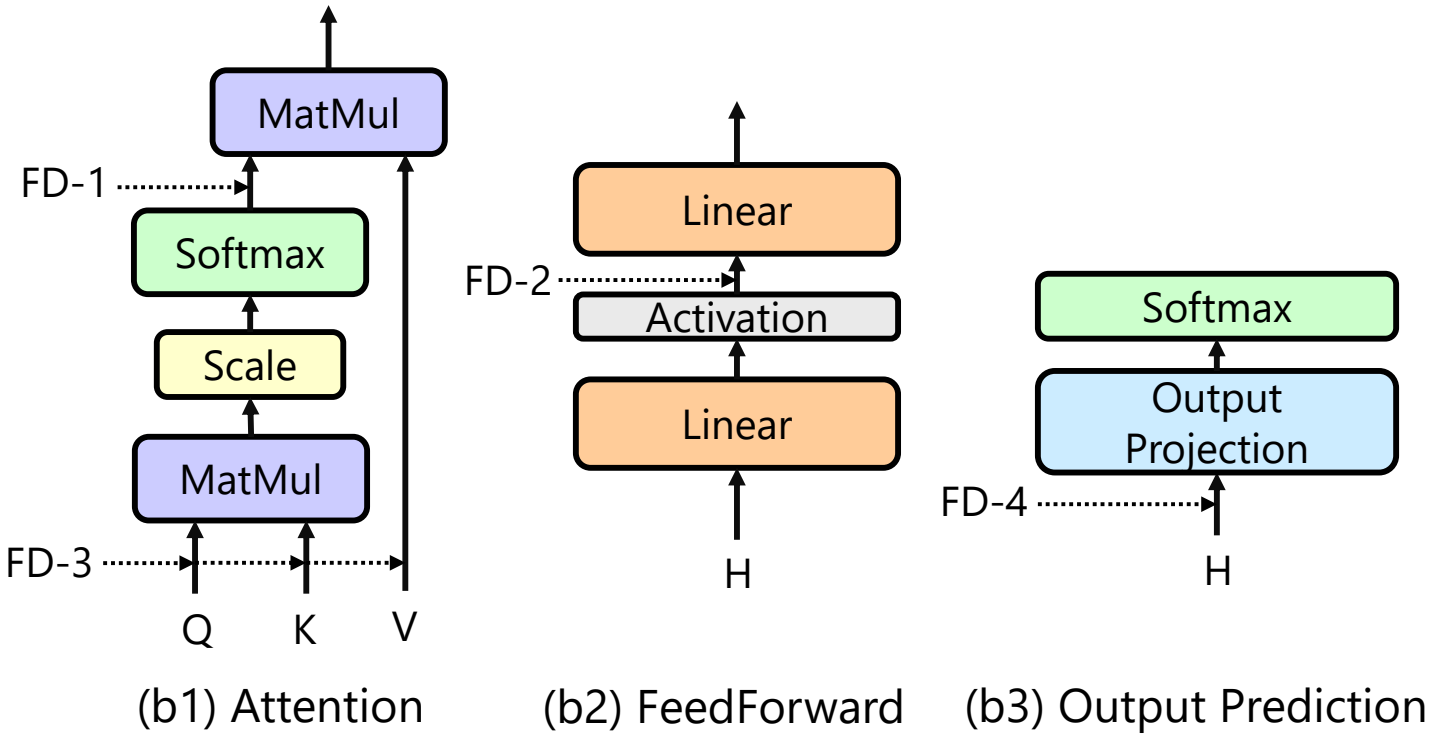


Figure 3: Structure and overview of feature dropout.

Structure Dropout & Data Dropout

Structure Dropout

- We adopt `LayerDrop` (Fan et al., 2020a) as the structure dropout, which drops some entire layers at training time and directly reduces the Transformer model size.

Data Dropout

- Direct data dropout brings the risk of losing high-quality training samples.
- We propose a two-stage data dropout strategy.

Two-stage data dropout strategy

Given a sequence, with probability p_k, we keep the original sequence and do not apply data dropout. If data dropout is applied, for each token, with another probability p, we will drop the token.

UniDrop Integration

- Theoretically demonstrate that the above three dropouts play different roles in preventing Transformer from overfitting.
- `UniDrop` finally unites feature dropout, LayerDrop, and the two-stage data dropout strategy into Transformer models.

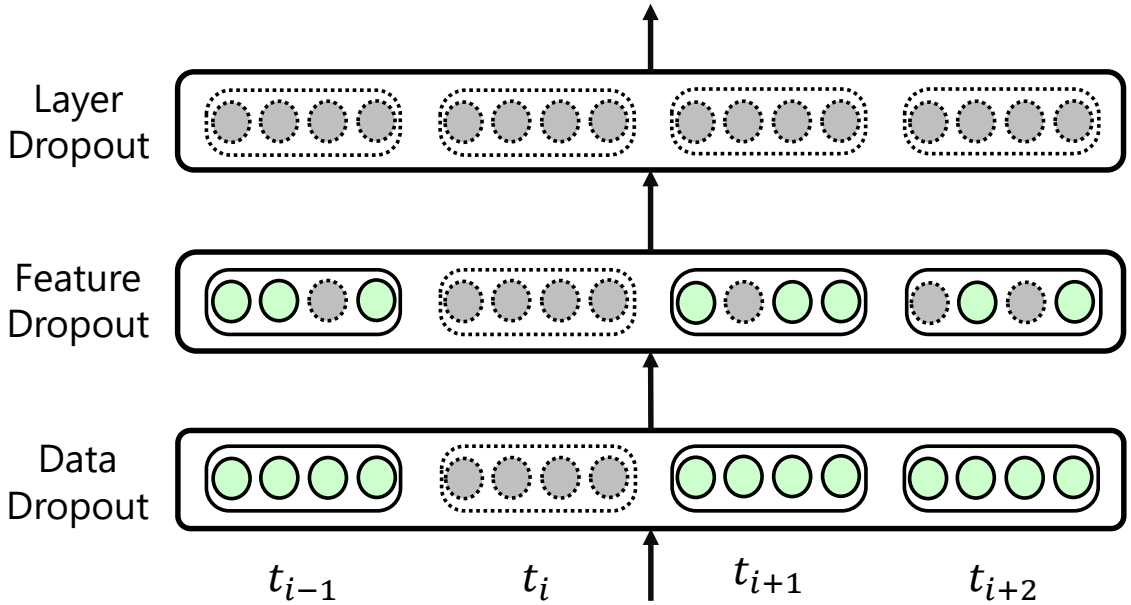


Figure 4: Different dropout components in UniDrop. The gray positions denote applying the corresponding dropout.

Outline

Datasets

- We conduct machine translation experiments on the widely acknowledged IWSLT14 datasets[1] with multiple language pairs, including English↔German (En↔De), English↔Romanian (En↔Ro), English↔Dutch (En↔Nl), and English↔Portuguese-Brazil (En↔Pt-br), a total number of 8 translation tasks.

Datasets	Train	Dev	Test
En↔De	160k	7k	7k
En↔Ro	180k	4.7k	1.1k
En↔Nl	170k	4.5k	1.1k
En↔Pt-br	175k	4.5k	1.2k

Table 2: Statistics for machine translation datasets.

[1] https://wit3.fbk.eu/mt.php?release=2014-01

Model Configuration

- We use the `transformer_iwslt_de_en` configuration[2] for all Transformer models.
- `UniDrop` configuration
  - FD rates: 0.1
  - SD rate: 0.1 (only applied on decoder)
  - DD rates: the sequence keep rate $p_k = 0.5$, token dropout rate $p = 0.2$

[2] `https://github.com/pytorch/fairseq`

Neural Machine Translation

Main Results

	En→De	De→En	En→Ro	Ro→En	En→Nl	Nl→En	Nn→Pt-br	Pt-br→En	Avg.	△
Transformer	28.67	34.84	24.74	32.14	29.64	33.28	39.08	43.63	33.25	-
+FD	29.61	36.08	25.45	33.12	30.37	34.50	40.10	44.74	34.24	+0.99
+SD	29.03	35.09	25.03	32.69	29.97	33.94	39.78	44.02	33.69	+0.44
+DD	28.83	35.26	24.98	32.76	29.72	34.00	39.50	43.71	33.59	+0.34
+`UniDrop`	**29.99**	**36.88**	**25.77**	**33.49**	**31.01**	**34.80**	**40.62**	**45.62**	**34.77**	+1.52
w/o FD	29.24	35.68	25.18	33.17	30.16	33.90	39.97	44.81	34.01	+0.76
w/o SD	29.92	36.70	25.59	33.26	30.55	34.75	40.45	45.60	34.60	+1.35
w/o DD	29.76	36.38	25.44	33.26	30.86	34.55	40.37	45.27	34.49	+1.24

Table 3: Machine translation results of different models on IWSLT14 translation datasets. Avg. and △ denote the average results of the 8 translation tasks and improvements compared with the standard Transformer. Best results are in bold.

Summary:

- Transformer+`UniDrop` achieves the most improvements across all translation tasks, which demonstrates the effectiveness of UniDrop for the Transformer.
- Ablation study validates the necessity of FD, SD, and DD for `UniDrop`.

Comparisons

Approaches	BLEU
Adversarial MLE (Wang et al., 2019b)	35.18
DynamicConv (Wu et al., 2019)	35.20
Macaron (Lu et al., 2019)	35.40
IOT (Zhu et al., 2021)	35.62
Soft Contextual Data Aug (Gao et al., 2019)	35.78
BERT-fused NMT (Zhu et al., 2020)	36.11
MAT (Fan et al., 2020b)	36.22
MixReps+co-teaching (Wu et al., 2020)	36.41
Transformer	34.84
+`UniDrop`	**36.88**

Table 4: Comparison with existing works on IWSLT-2014 De→En translation task.

Summary:

- Transformer+`UniDrop` with dropout only outperforms the compared works including the training algorithm design (Wang et al., 2019b), model architecture design (Lu et al., 2019; Wu et al., 2019), and even BERT-fused model (Zhu et al., 2020).

Text Classification

Datasets

- We evaluate different methods on two groups of text classification datasets. The first group is from GLUE tasks (Wang et al., 2019a). The second group is some widely used text classification datasets in previous works (Voorhees and Tice, 1999; Maas et al., 2011; Zhang et al., 2015).

Datasets	Classes	Train	Dev
MNLI	3	393k	20k
QNLI	2	105k	5.5k
SST-2	2	67k	0.9k
MRPC	2	3.7k	0.4k
Datasets	Classes	Train	Test
IMDB	2	25k	25k
Yelp	5	650k	50k
AG's News	4	120k	76k
TREC	6	5.4k	0.5k

Table 5: Statistics for text classification datasets.

Text Classification

Main Results

	MNLI	QNLI	SST-2	MRPC
BiLSTM+Attn, CoVe	67.9	72.5	89.2	72.8
BiLSTM+Attn, ELMo	72.4	75.2	91.5	71.1
$\text{BERT}_{\text{BASE}}$	84.4	88.4	92.9	86.7
$\text{BERT}_{\text{LARGE}}$	86.6	92.3	93.2	88.0
$\text{RoBERTa}_{\text{BASE}}$	87.1	92.7	94.7	89.0
+UniDrop	**87.8**	**93.2**	**95.5**	**90.4**

Table 6: Accuracy on GLUE tasks (dev set).

	IMDB	Yelp	AG	TREC
Char-level CNN	-	62.05	90.49	-
VDCNN	-	64.72	91.33	-
DPCNN	-	69.42	93.13	-
ULMFiT	95.40	-	94.99	96.40
$\text{BERT}_{\text{BASE}}$	94.60	69.94	94.75	97.20
$\text{RoBERTa}_{\text{BASE}}$	95.7	70.9	95.1	97.6
+UniDrop	**96.0**	**71.4**	**95.5**	**98.0**

Table 7: Accuracy on the typical text classification datasets.

Summary:

- UniDrop further improves the performance of Transformer models even using strong $\text{RoBERTa}_{\text{BASE}}$ as backbone.

Outline

Analysis

Overfitting

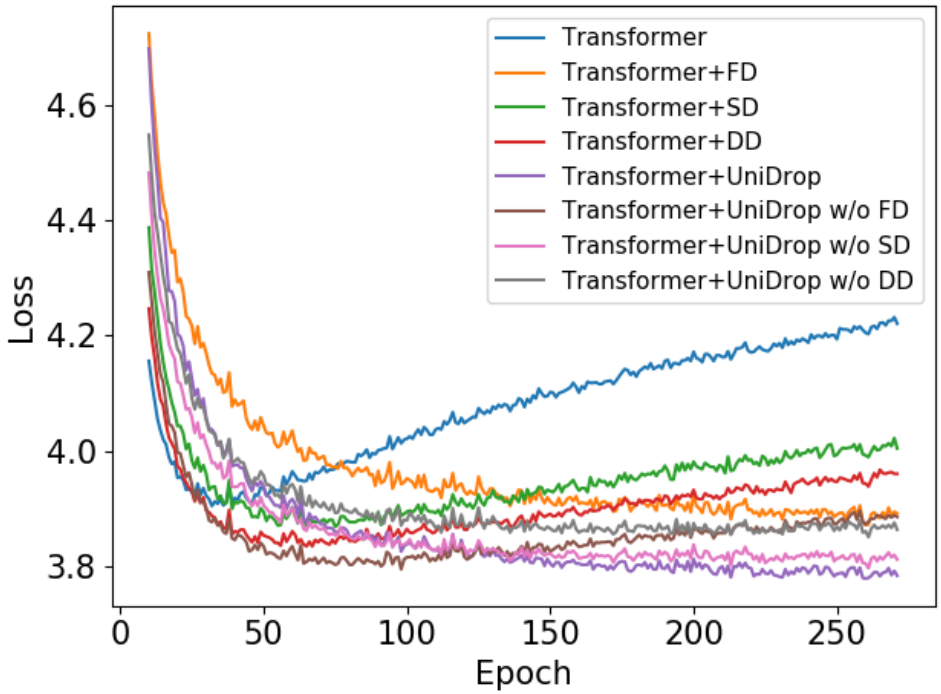


Figure 5: The dev loss of different models on IWSLT14 De→En translation task.

Summary:

- The standard Transformer is quickly overfitted during training.
- Transformer+`UniDrop` achieves the lowest dev loss and shows great advantage to prevent Transformer from overfitting.

Analysis

Ablation Study

	De→En	En→De	Ro→En
Transformer	34.84	28.67	32.14
+UniDrop	36.88	29.99	33.49
w/o FD-1	36.72	29.84	33.33
w/o FD-2	36.57	29.76	33.28
w/o FD-3	36.59	29.83	33.31
w/o FD-4	36.65	29.59	33.24
w/o 2-stage DD	36.61	29.78	33.12

Table 8: Ablation study of data dropout and different feature dropouts on IWSLT14 De→En, En→De, and Ro→En translation tasks.

Summary:

- In multi-head attention module, FD-3 brings more improvement than FD-1, showing the insufficiency of only applying FD-1 for the Transformer.
- The model without 2-stage DD has the lower BLEU scores, indicating the necessity of keeping the original sequence for data dropout.

Analysis

Effects of Different Dropout Rates

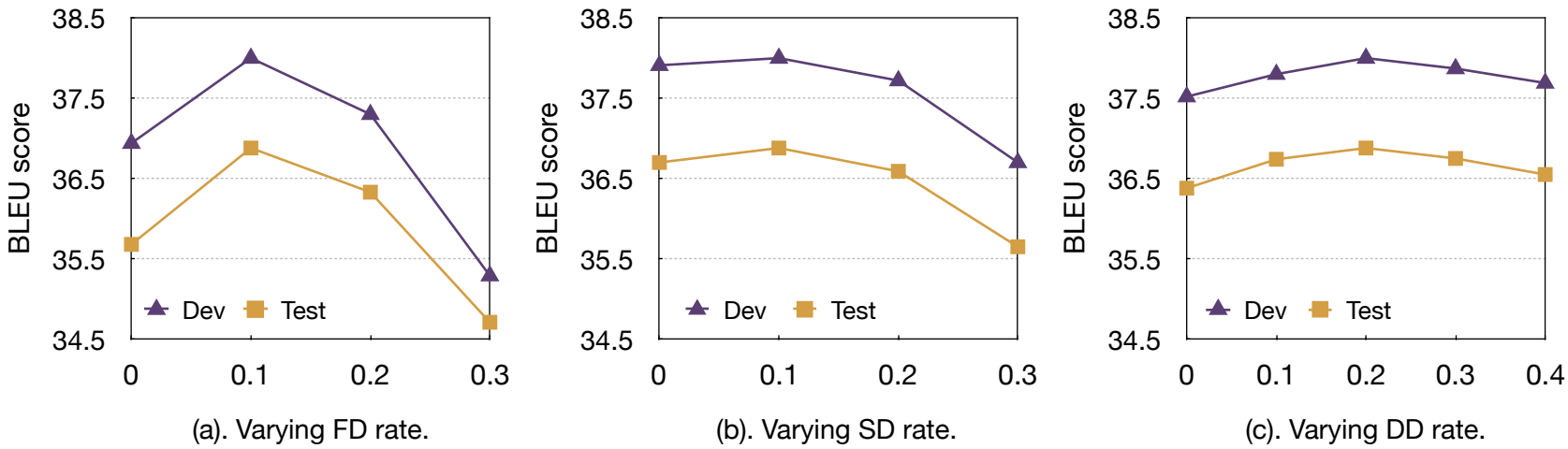


Figure 6: The BLEU scores of Transformer+`UniDrop` on IWSLT14 De→En translation dev set and test test, with varying the rates of FD, SD and DD respectively.

Summary:

- The performance first increases then decreases when varying the dropout rates from small to large.
- Change of FD dropout rate makes the most significant impact on the model performance since FD contains four feature dropout positions.

Outline

Conclusions

- We introduce `UniDrop` to unite three different level dropout techniques, i.e., feature dropout, structure dropout, and data dropout, into a robust one for Transformer.
- We theoretically demonstrate that the three dropouts play different roles in regularizing Transformer model and improving the robustness of the model.
- Extensive results indicate that Transformer models with UniDrop can achieve strong or even SOTA performances on sequence generation and classification tasks.

End.

Thanks!

Angela Fan, Edouard Grave, and Armand Joulin. 2020a. Reducing transformer depth on demand with structured dropout. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Yang Fan, Shufang Xie, Yingce Xia, Lijun Wu, Tao Qin, Xiang-Yang Li, and Tie-Yan Liu. 2020b. Multi-branch attentive transformer. *CoRR*, abs/2006.10270.

Fei Gao, Jinhua Zhu, Lijun Wu, Yingce Xia, Tao Qin, Xueqi Cheng, Wengang Zhou, and Tie-Yan Liu. 2019. Soft contextual data augmentation for neural machine translation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5539–5544.

Yiping Lu, Zhuohan Li, Di He, Zhiqing Sun, Bin Dong, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Understanding and improving transformer from a multi-particle dynamic system point of view. *arXiv preprint arXiv:1906.02762*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *The 49th Annual Meeting of the Association for Computational*

Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA, pages 142–150. The Association for Computer Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Ellen M. Voorhees and Dawn M. Tice. 1999. The TREC-8 question answering track evaluation. In *Proceedings of The Eighth Text REtrieval Conference, TREC 1999, Gaithersburg, Maryland, USA, November 17-19, 1999*, volume 500-246 of *NIST Special Publication*. National Institute of Standards and Technology (NIST).

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Dilin Wang, ChengYue Gong, and Qiang Liu. 2019b. Improving neural language modeling via adversarial training. In *Proceedings of the 36th*

International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, volume 97 of *Proceedings of Machine Learning Research*, pages 6555–6565. PMLR.

Felix Wu, Angela Fan, Alexei Baevski, Yann Dauphin, and Michael Auli. 2019. Pay less attention with lightweight and dynamic convolutions. In *International Conference on Learning Representations*.

Lijun Wu, Shufang Xie, Yingce Xia, Yang Fan, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2020. Sequence generation with mixed representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015*, pages 649–657.

Jinhua Zhu, Lijun Wu, Yingce Xia, Shufang Xie, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. 2021. {IOT}: Instance-wise layer reordering for transformer structures. In *International Conference on Learning Representations*.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. 2020. Incorporating BERT into neural machine translation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.