(スライド作成途中) Intrinsic Dimensionality Explains the Effectiveness of Language Model Fine-Tuning

第13回 最先端NLP勉強会

2021年 9月 16-17日

読んだ人：篠田一聡 (東大 相澤研)

書誌情報

Intrinsic Dimensionality Explains the Effectiveness of Language Model Fine-Tuning

Armen Aghajanyan
Facebook AI
armenag@fb.com

Sonal Gupta
Facebook
sonalgupta@fb.com

Luke Zettlemoyer
Facebook AI
University of Washington
lsz@fb.com

ACL2021 long
Outstanding paper

導入

事前学習済み言語モデルのFine-tuningはなぜうまくいくのか？

- BERT等の事前学習済み言語モデルを何らかのターゲットタスクでfine-tuningすることが、ほとんどのNLPタスクで行われている。

- しかし、なぜターゲットタスクのデータが少ないのにfine-tuningがうまく行くのかまだよく分かっていない。
  - モデルのパラメータ数は数億で、なぜか多い方が精度がいい
  - 一方でターゲットタスクの訓練データ数は数百〜数千
  - 単純な勾配降下法で十分に学習が行えて高い精度を出せる

Intrinsic dimensionalityによる分析

貢献

- Intrinsic dimensionality (Li et al., 2018) を使って事前学習済み言語モデルを分析することでこんなことが分かった
  - Fine-tuning 中に数百〜数千程度のパラメータをいじれば、モデルをフルに使った時の9割の精度を達成できることがわかった
  - MDLとしてみると、、、
  - パラメータの多い大きなモデルほど、 Intrinsic dimensionalityが少ないことがわかった
  - 言語モデルの事前学習は、このIntrinsic dimensionalityを減らす効果があった
  - 汎化誤差

Intrinsic dimensionalityとは

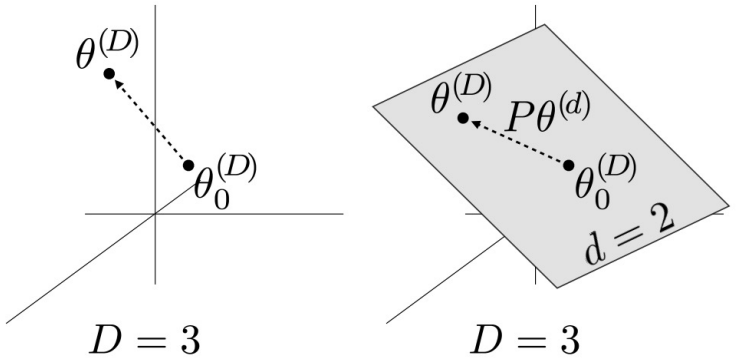
- Intrinsic dimensionalityは、精度や指標がある一定の値をとるのに必要な最小のパラメータ数

$$\theta^D = \theta_0^D + P(\theta^d)$$

θ_0^D: モデルの初期パラメータ (D次元空間の１点)
θ^D: モデルの学習後のパラメータ
θ^d: d次元空間のベクトル

$$P : \mathbb{R}^d \rightarrow \mathbb{R}^D$$

P: d次元空間の点(ベクトル)をD次元空間の点(ベクトル)に変換する関数



(出典: Li et al., 2018)

この例では2(=d)次元ベクトルをPで3(=D)次元に変換している。

- 原理的には、モデル、データセット、指標の値、 P が決まると、Intrinsic dimensionalityが決まる

この研究での Intrinsic dimensionalityの求め方

何を求めるか？

- パラメータ数Dのモデルを使って85%の精度を達成できるなら、その9割の精度 (85%*0.9 = 76.5%) を達成するのに必要な最小のパラメータ数を、d_{90}と示す。

どうやって求めるか？

- 手順
  1. 動かせるパラメータ数がDの時の精度αを求める。
  2. dを適当に決める→SGDで最適なθ^dを探す→精度β_dを求める
  3. 余裕があればいろいろなdで 2. を試す OR 余裕がなければ二分探索
  4. 試したdの中で、精度β_dがαの9割を超えているdのうち最小のdがd_{90}

Pをどうやって計算するか(1/2)

- Li et al. (2018)が提案したもののうち、計算コストの削減を優先して、Fastfood transform (Le et al., 2013) (以下M) を採用

$$\theta^D = \theta_0^D + \theta^d M \qquad M = HG\Pi HB$$

  - H: Hadamard matrix
  - G: random diagonal matrix with independent standard normal entries
  - B: random diagonal matrix with equal probability ± 1 entries
  - Π: random permutation matrix
- 要は掛け算の計算コストが低い5つの行列をつなげただけ。
- 訓練中、動かせるパラメータはθ^dのみで、他 (M含む) は固定
- Direct Intrinsic Dimension (**DID**) と呼ぶことにする。

Pをどうやって計算するか(2/2)

- Layerを考慮してIntrinsic dimensionを計算した方がモデルの構造を考慮できて良い

$$\theta_i^D = \theta_{0,i}^D + \lambda_i P(\theta^{d-m})_i$$

  - Layerごとに係数λ_iをかける。
  - 動かせるパラメータは、θ^{d-m} と λ_1~λ_mのd個
  - Structure-Aware Intrinsic Dimension (**SAID**) と呼ぶことにする。

実験

実験1: NLPタスクでのIntrinsic Dimensionality

パラフレーズか否かを予測する以下の２つのタスクでd_{90}を求める。

- MRPC（訓練データ：3700）
- QQP（訓練データ：363k）

各データセット・各モデルについて、 learning rate を4種類、d は10から10000の間の100種類を試して、9割の精度に達した最小のdがintrinsic dimension。

実験1: NLPタスクでのIntrinsic Dimensionality

わかったこと：

- **全体的にモデルのパラメータ数D（数億）に比べてかなり少ないパラメータ（数百～数千）を学習するだけで元の9割の精度を達成できる。**
- d_{90} はRoBERTa-Large < RoBERTa-Base < BERT-Large < BERT-Base
- 一貫して、SAID < DID … モデルの構造を考慮することでd_{90}を減らせる。

考察：

- P: d→Dの決め方がランダムでかなり雑であったことを考えると、真のd_{90}はもっと小さいことが予想される。

	SAID		DID	
Model	MRPC	QQP	MRPC	QQP
BERT-Base	1608	8030	1861	9295
BERT-Large	1037	1200	2493	1389
RoBERTa-Base	896	896	1000	1389
RoBERTa-Large	**207**	**774**	322	**774**

Table 1: Estimated d_{90} intrinsic dimension computed with SAID and DID for a set of sentence prediction tasks and common pre-trained models.

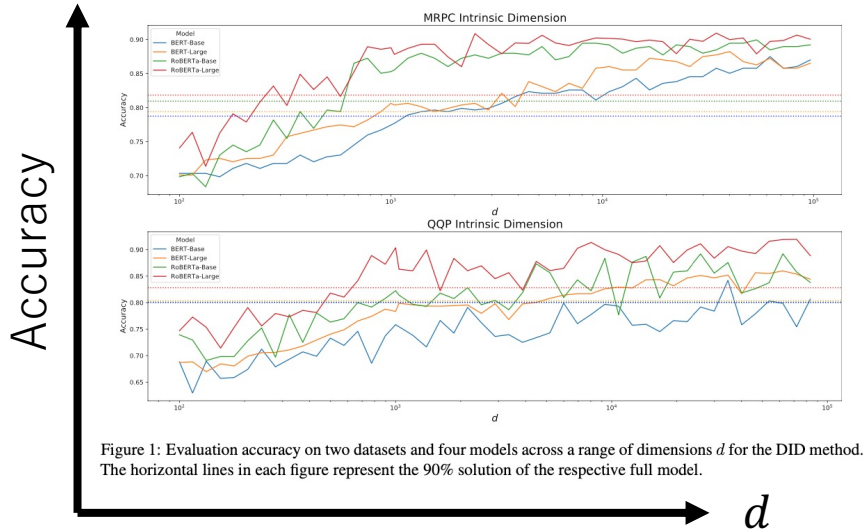


Figure 1: Evaluation accuracy on two datasets and four models across a range of dimensions d for the DID method. The horizontal lines in each figure represent the 90% solution of the respective full model.

実験2: Pre-Training and Intrinsic Dimensionality

- 言語モデルの事前学習によってIntrinsic dimensionが小さくなっているのではないか？
  - → RoBERTa-Baseをスクラッチから訓練して、訓練中のチェックポイントのintrinsic dimensionalityを計算

実験2: Pre-Training and Intrinsic Dimensionality

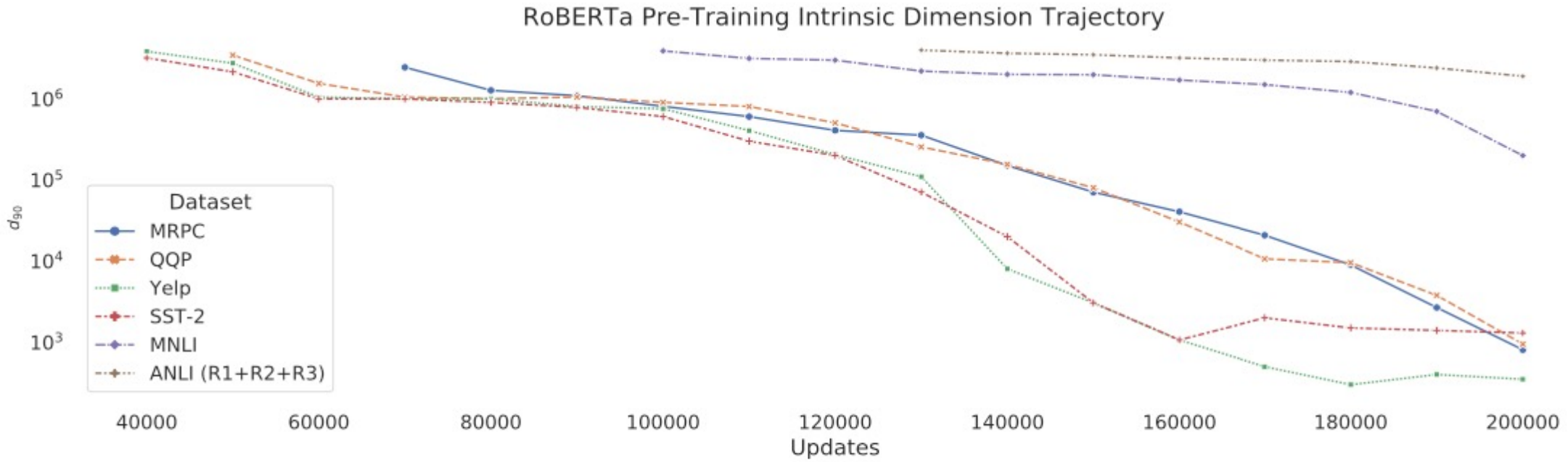


Figure 2: Every 10k updates of RoBERTa-Base that we trained from scratch, we compute d_{90} for six datasets; MRPC, QQP, Yelp Polarity, SST-2, MNLI, and ANLI. If we were unable to compute a d_{90} for a specific checkpoint, we do not plot the point, hence some datasets start at later points. Unable to compute means either we could not fine-tune the full checkpoint to accuracy above majority class or stabilize SAID training.

実験2: Pre-Training and Intrinsic Dimensionality

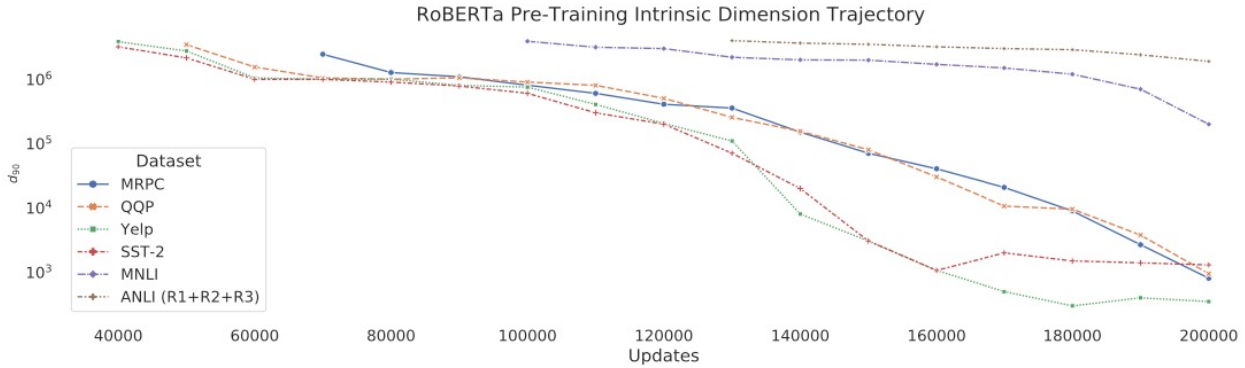


Figure 2: Every 10k updates of RoBERTa-Base that we trained from scratch, we compute d_{90} for six datasets; MRPC, QQP, Yelp Polarity, SST-2, MNLI, and ANLI. If we were unable to compute a d_{90} for a specific checkpoint, we do not plot the point, hence some datasets start at later points. Unable to compute means either we could not fine-tune the full checkpoint to accuracy above majority class or stabilize SAID training.

わかったこと：

- **ダウンストリームのデータセットにアクセスできないのにも関わらず、事前学習が進むほど、どのタスクでもIntrinsic dimension が減少していく**
- 難しいダウンストリームタスク(ANLIなど)ほど、 Intrinsic dimension が大きい⇒汎化と関係がある

実験3: Parameter Count and Intrinsic Dimension

- モデルのパラメータ数とIntrinsic dimensionには関係があるのではないか？
  - →様々な事前学習済み言語モデルでMRPCデータセットでのIntrinsic dimensionを計算した

実験3: Parameter Count and Intrinsic Dimension

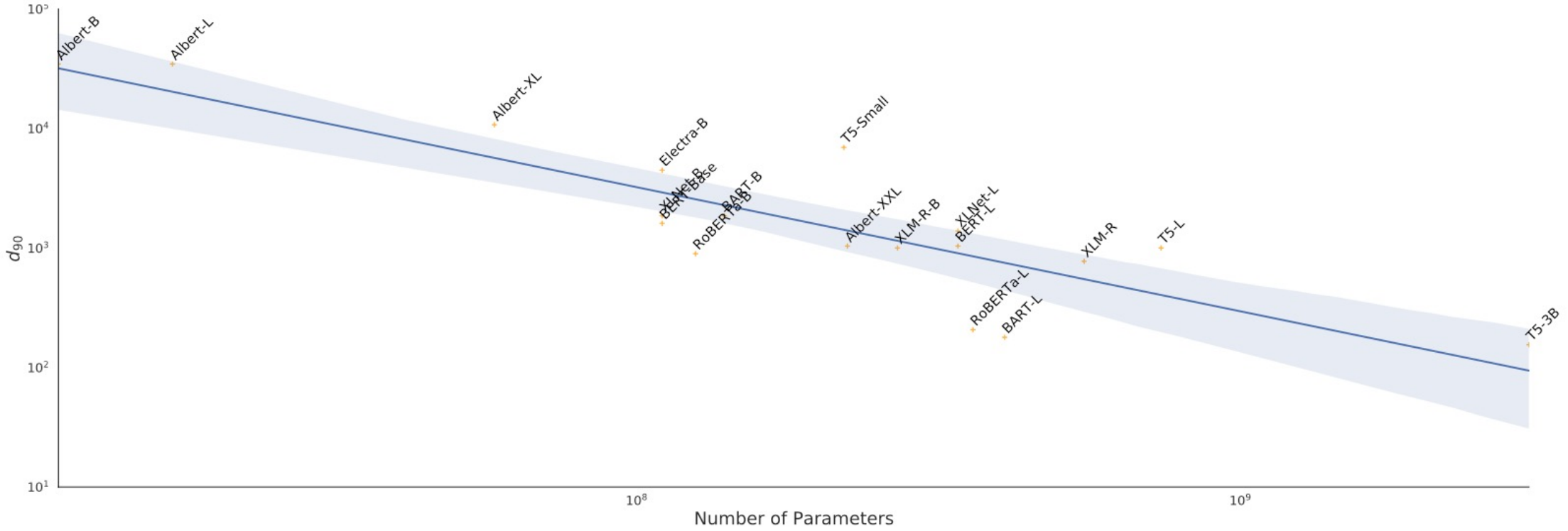


Figure 3: We calculate the intrinsic dimension for a large set of pre-trained models using the SAID method on the MRPC dataset.

実験3: Parameter Count and Intrinsic Dimension

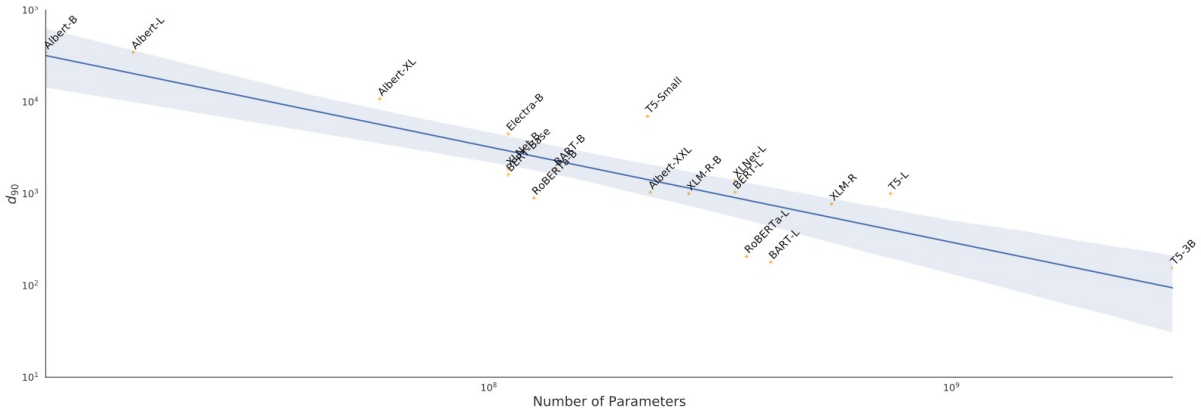


Figure 3: We calculate the intrinsic dimension for a large set of pre-trained models using the SAID method on the MRPC dataset.

わかったこと

- **モデルのパラメータ数が多いほど、Intrinsic dimensionが減少する傾向**
- 違うデータセットでも試したが同様の傾向が見られた
- モデルのパラメータ数が同程度の場合は、事前学習方法が重要になる。

実験4: Generalization Bounds through Intrinsic Dimension

- 実験2で事前学習が進むほどIntrinsic dimensionが減少したことから、 Intrinsic dimensionが減少すれば汎化性能も向上するのではないか？
  - → RoBERTaの事前学習中のチェックポイントを使って実験

実験4: Generalization Bounds through Intrinsic Dimension

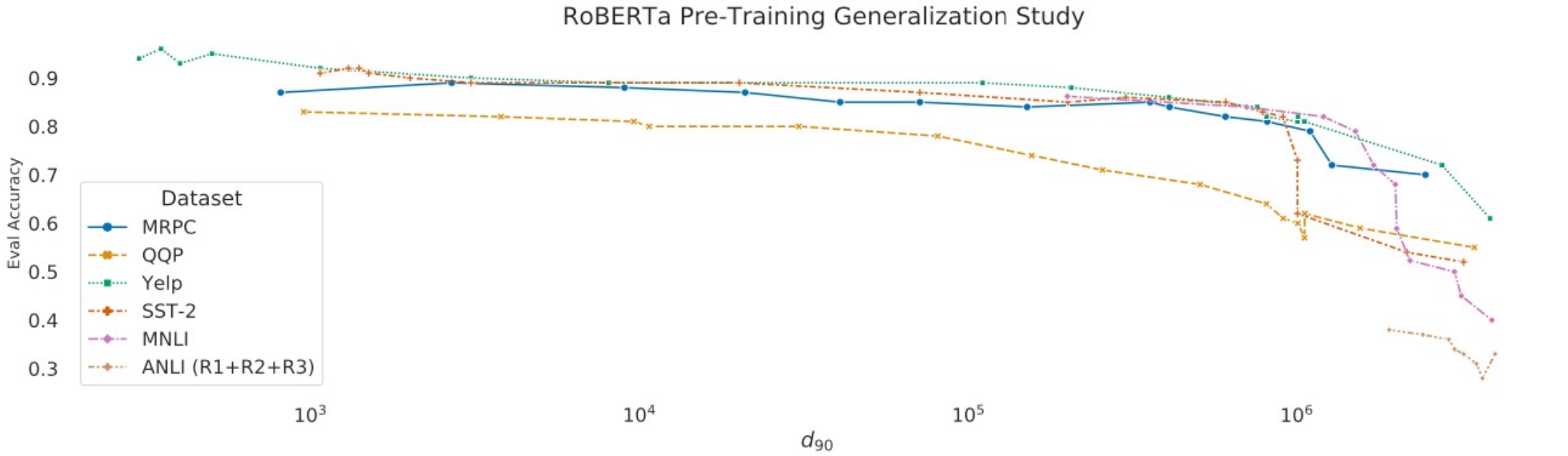


Figure 4: The evaluation accuracy of six datasets across various intrinsic dimensionalities. There is a strong general trend that pre-trained models that are able to attain lower intrinsic dimensions generalize better.

結果

- **Intrinsic dimensionが小さいほど、ダウンストリームタスクでの精度が高い**

実験4: Generalization Bounds through Intrinsic Dimension

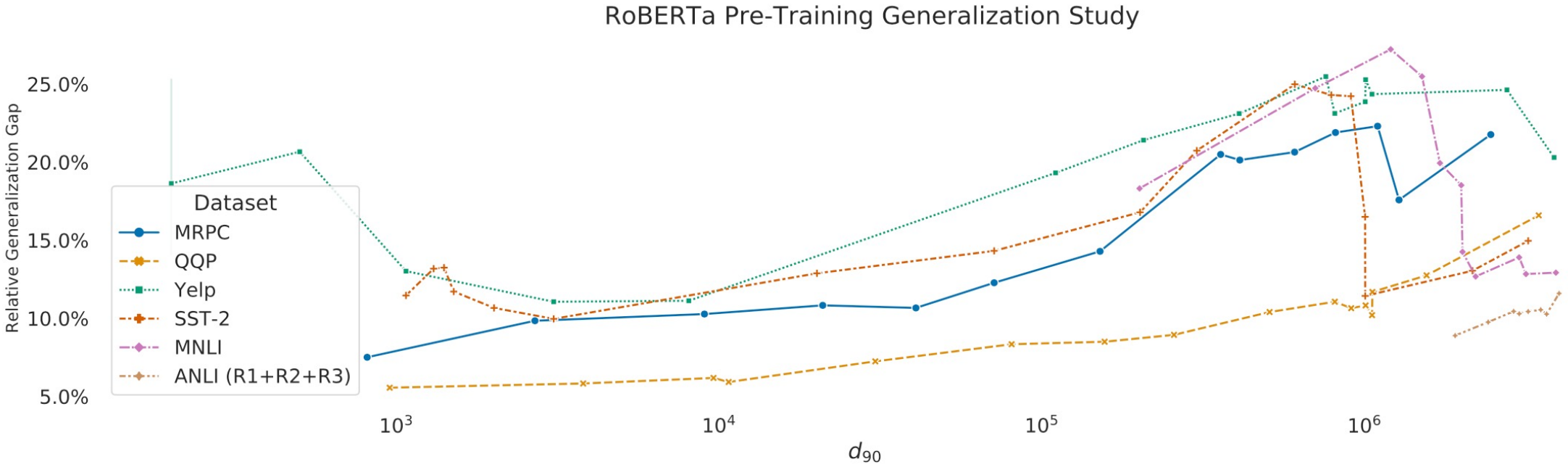


Figure 5: The intrinsic dimension and the respective relative generalization gap across a set of varied tasks.

結果

- **Intrinsic dimensionが小さいほど、relative generalization gapが小さい**

※Relative generalization gap $= \frac{acc_{train} - acc_{eval}}{1 - acc_{eval}}$

理論的な裏付け

Generalization Bounds

- Intrinsic dimension と 汎化の関係について理論的な裏付けを行った。

Conclusion

References

- Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. 2018. Measuring the intrinsic dimension of objective landscapes. *arXiv preprint arXiv:1804.08838*.
- Quoc Le, Tama ´s Sarlo ´s, and Alex Smola. 2013. Fastfood-approximating kernel expansions in loglinear time. In *Proceedings of the international conference on machine learning*, volume 85.