It’s Not Just Size That Matters:

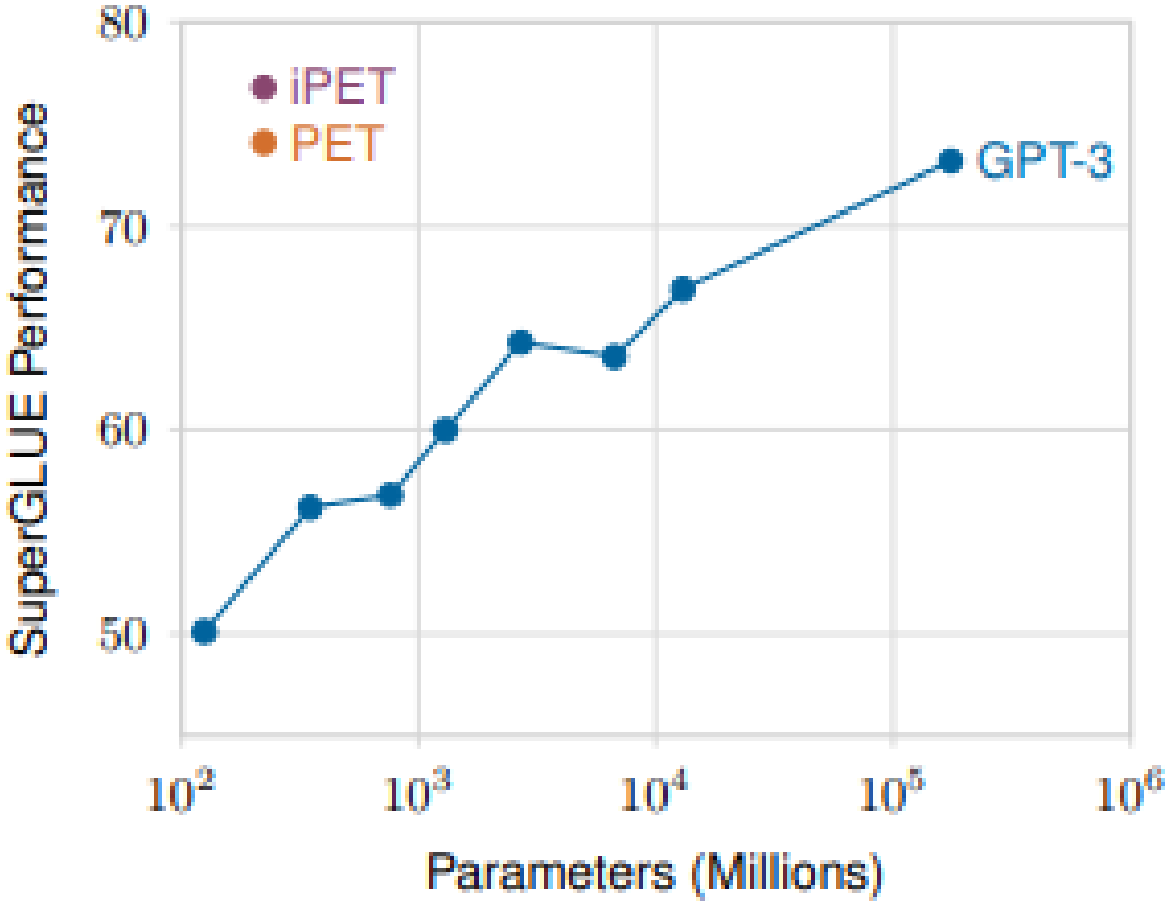
Small Language Models Are Also Few-Shot Learners

2020/10/10 @dl-study

背景

- **Language Models are Few-Shot Learners(2020)**
  - ➢パラメータ数1750億の巨大モデル「GPT-3」を提案
  - ➢Fine-tuning を使わないでSOTAに近い結果を残した。

まず性能みてみよう



縦軸：SuperGLUEのスコア
横軸：パラメータ数(1億~1750億)

GPT-3の論文ではパラメータ数を変えて実験しているので点がいっぱいある

本論文で提案しているのがPET/iPET。
信じられない場所に点があるな…

この論文、このように進みます

- PETという**手法**を紹介する。
- ALBERTにPETを取り入れる。
  ※ALBERT：BERTの軽量モデル
- GPT-3と比べる。

GPT-3の欠点

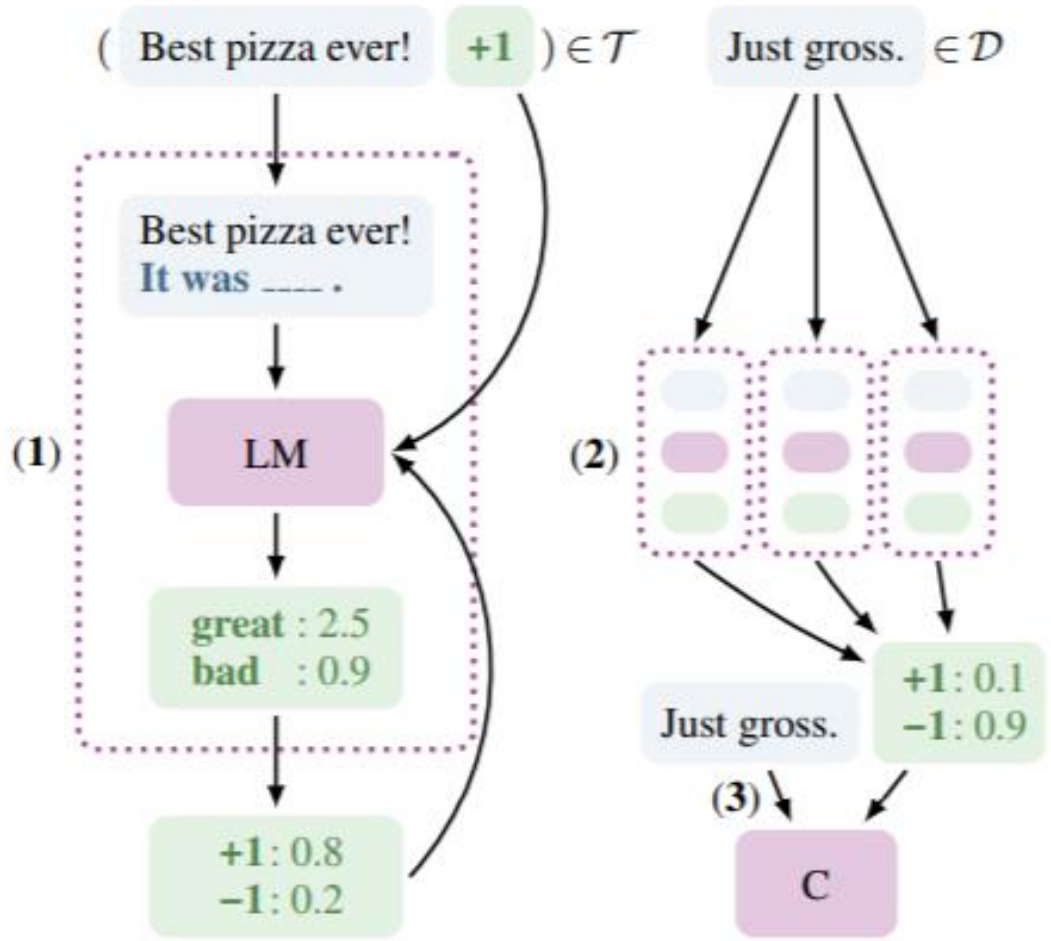
- モデルが大きすぎて現実的に使えない（全否定）

- 入力サイズが限られているので、入力できるexampleの数が限られている。

PETの紹介

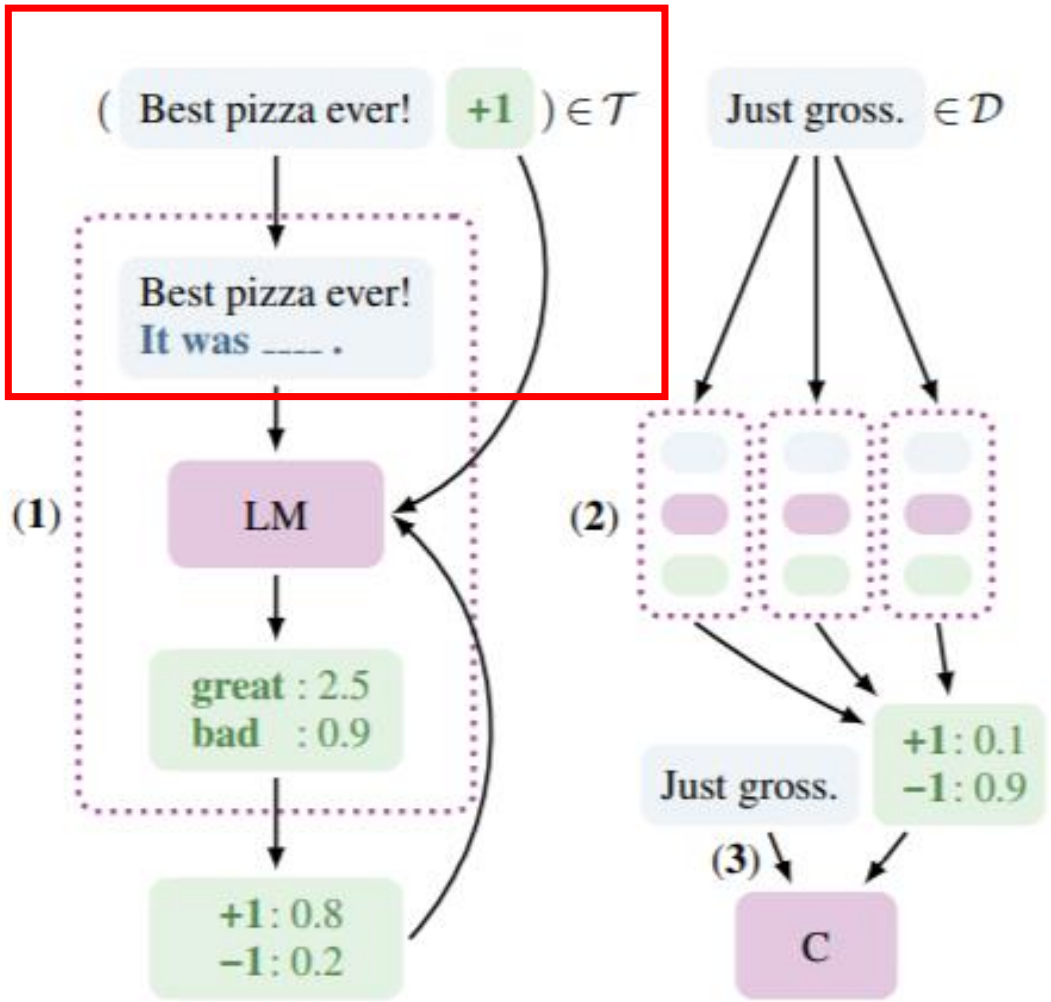
- PET(Pattern-Exploiting Training)
- PETの登場は別論文

PET モデル概要



- レビューの極性（ポジティブ/ネガティブ）分析を行っている
- データ $\mathcal{T}$（ラベルあり）を使って学習し、**データ $\mathcal{D}$（ラベルなし）にラベリングすることが目標。**
- 左は学習時, 右は実行時
- ※Just gross：「気持ち悪い」という意味のスラング

PET 学習①



- 関数 $\mathcal{P}$を使って、入力 x を変換
- “____” = [Mask]
- 文章→極性予測 のタスクから、**[Mask]に入る単語の予測タスクになる**
- **変換の仕方とマスクに入る単語の候補は人が決めてる・タスクごとに違う。**
  たぶん単純なやり方で変換していると思います

PET 学習①

タスク：レビューの極性分析

$P_1(a) =$ It was ____. a

$P_2(a) =$ a. All in all, it was ____.

$P_3(a) =$ Just ____! $\|$ a

$P_4(a) =$ $a \,\|$ In summary, the restaurant is ____.

$v(1) =$ terrible $\quad v(2) =$ bad $\quad v(3) =$ okay

$v(4) =$ good $\quad v(5) =$ great

PET 学習①

タスク：ニュース記事をカテゴリ別に分類する

$P_1(\mathbf{x}) =$ ____: a b　　$P_2(\mathbf{x}) =$ a (____) b

$P_3(\mathbf{x}) =$ ____ – a b　　$P_4(\mathbf{x}) =$ a b (____)

$P_5(\mathbf{x}) =$ ____ News: a b

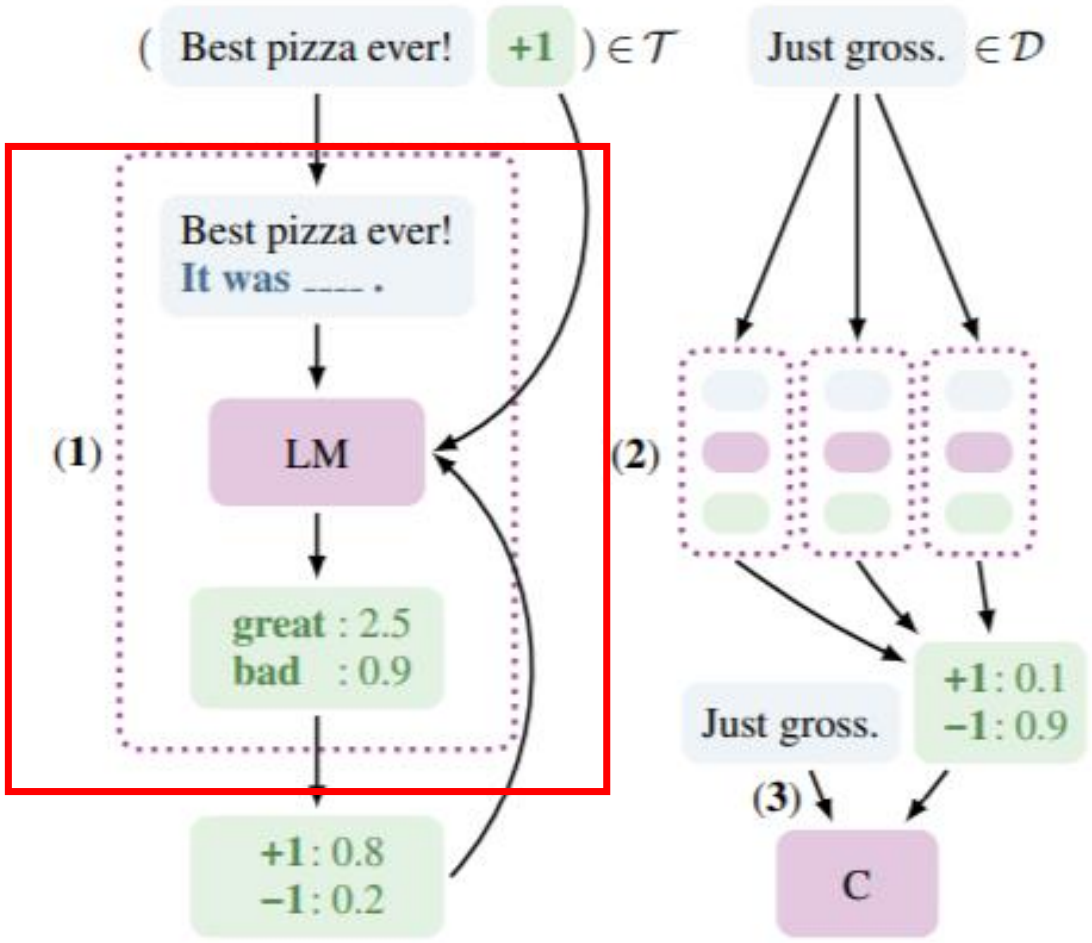
$P_6(\mathbf{x}) =$ [Category: ____] a b

$v(1) =$ World　　$v(2) =$ Sports

$v(3) =$ Business　　$v(4) =$ Tech

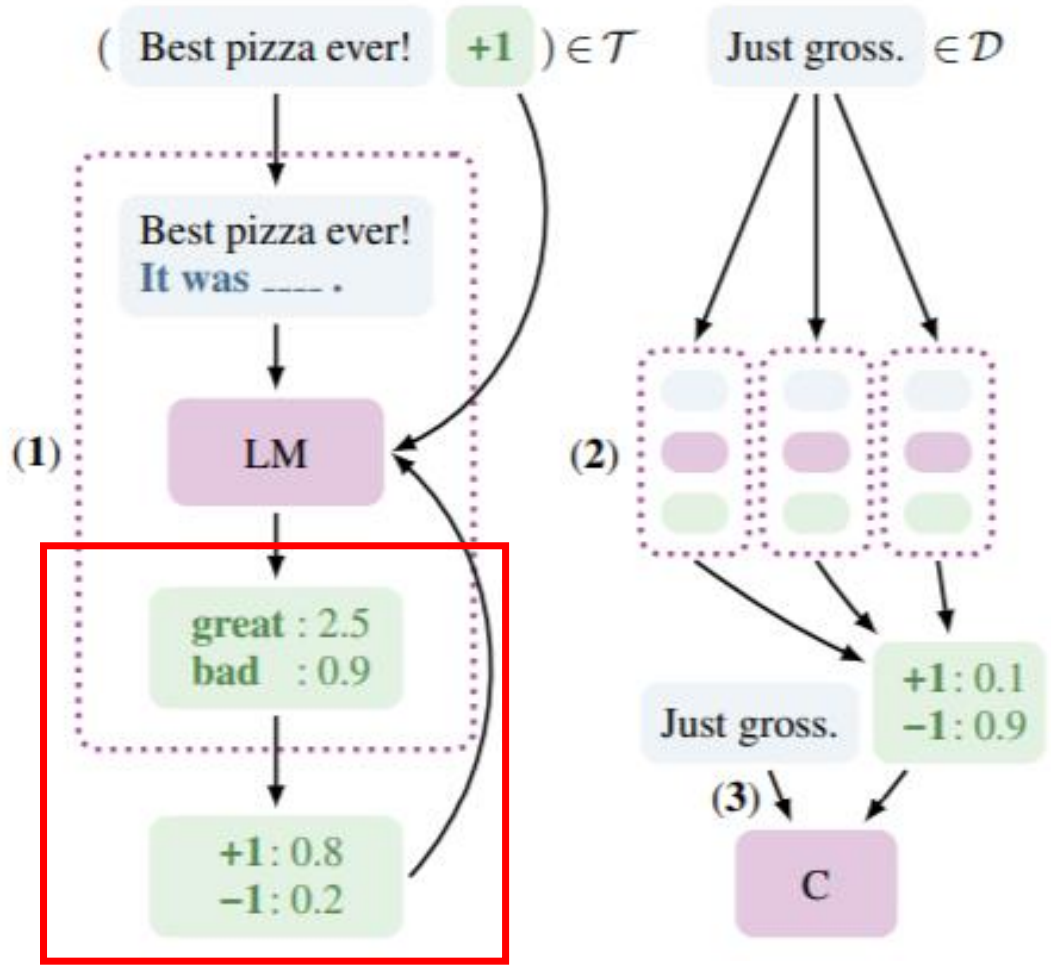
※a：見出し, b：本文

PET 学習②



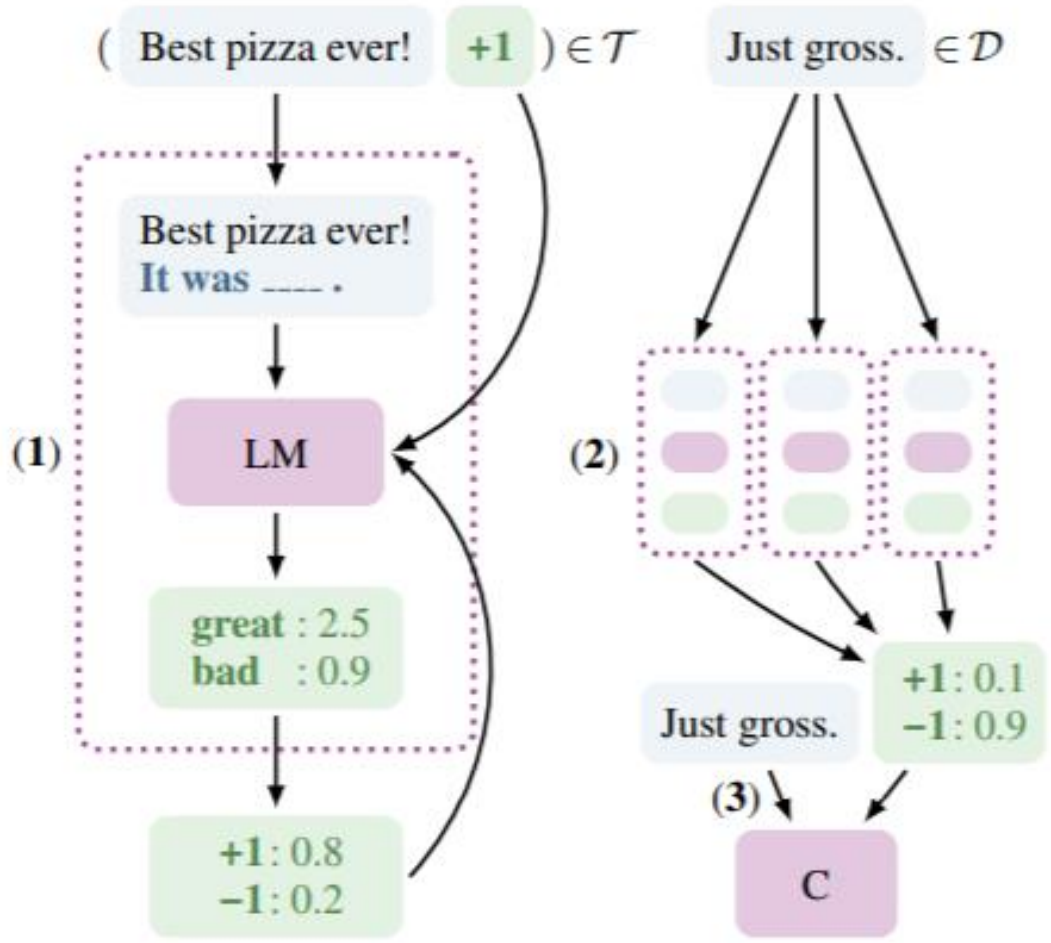
- Masked Language model（事前学習済み）を使って[Mask]を推測する
- [Mask]が一つでないといけない弱点があるが、いろんなタスクが同一のタスクとみなせるようになり、よさげ
- （図示されてないけど）関数 v がLMの推測結果を単語に変換する

PET 学習③



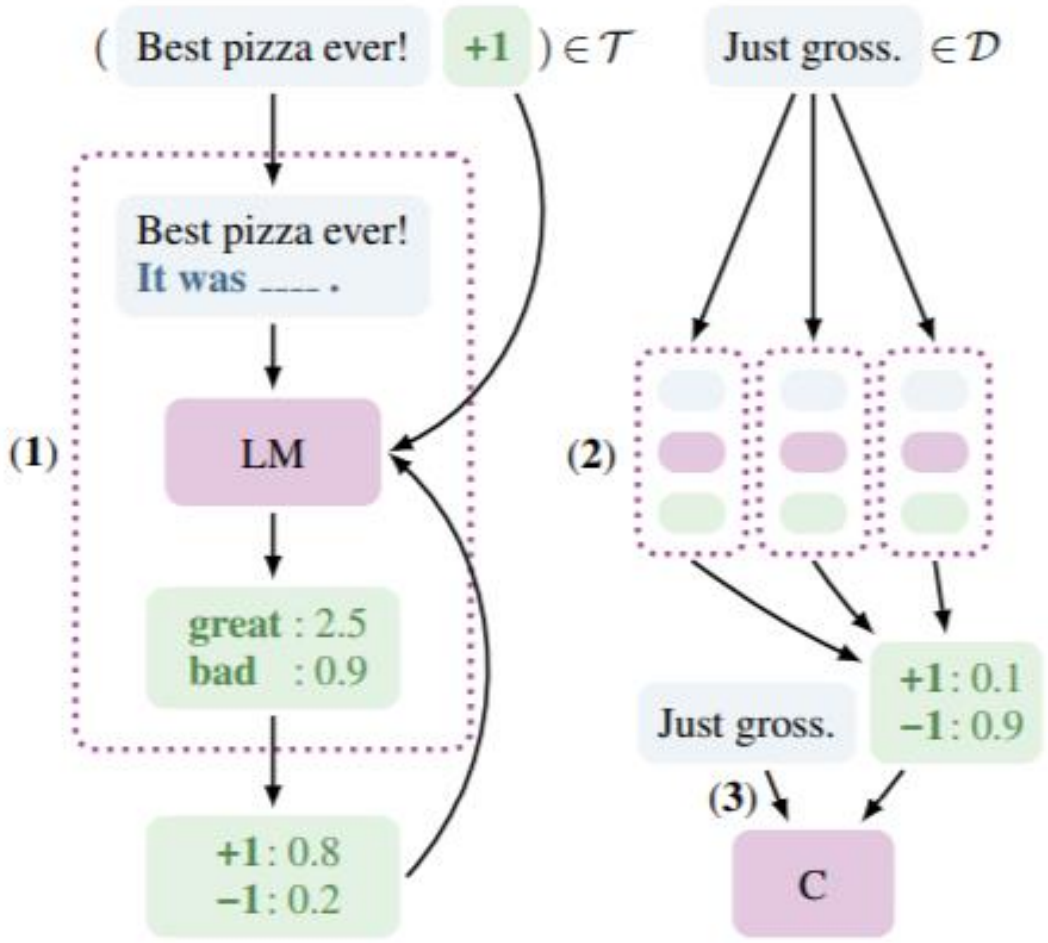
- Softmaxする（推測結果：q）
- 正解ラベルと q の交差エントロピー誤差を取り、LMを更新
- これは T に対するfine-tuningであると論文中でも言ってるけど、fine-tuningありなの？って少し思ってしまった。

PET 学習(さいご)



- 点線で囲まれた部分をPVP(Pattern-verbalizer pair)とよぶ
- $\mathcal{T}$ が小さくてPVP１つでは不十分なので、**複数のPVPモデルを学習させておく。**

PET 推論



- 学習したPVPモデルを複数使ってデータ $\mathcal{D}$（ラベルなし）をラベリングする
- 推論結果は100%にはならないので、**Soft-label**とよぶ。

PET 推論

$$s_{\mathcal{M}}(l \mid \mathbf{x}) = \frac{1}{Z} \sum_{\mathbf{p} \in \mathcal{P}} w(\mathbf{p}) \cdot s_{\mathbf{p}}(l \mid \mathbf{x})$$

- $\mathcal{M}$：PVPモデルの集合
- $w(\rho)$：各PVPモデルの重み
- $s_\rho(l \mid x)$：入力 x としてラベル l に対するスコア
- $Z = \sum_{\rho \in \mathcal{P}} w(\rho)$

PET 推論

$$s_{\mathcal{M}}(l \mid \mathbf{x}) = \frac{1}{Z} \sum_{\mathbf{p} \in \mathcal{P}} w(\mathbf{p}) \cdot s_{\mathbf{p}}(l \mid \mathbf{x})$$

- $\mathcal{M}$：PVPモデルの集合
- $w(\rho)$：各PVPモデルの重み
- $s_\rho(l \mid x)$：入力 x としてラベル l に対するスコア
- $Z = \sum_{\rho \in \mathcal{P}} w(\rho)$

2つのパターン（②のほうがうまくいく）

①すべての ρ に $w(\rho) = 1$ を設定する

②fine-tuning前の ρ で推論した精度に応じて重みづけ

PET 推論

- $s_{\mathcal{M}}(l \mid x)$ にSoftmaxする→qを得る。
- (x, q) をSoft-labeled データセット $\mathcal{T}_c$ に追加する
- $\mathcal{T}_c$ を使ってモデルを学習（たぶん別のモデルだと思う）

iPET

- Iterative(反復)PET

[手法]

① PETのいくつかの ρ を選び、ラベルなしデータにSoft-labelingする

② データの中で自信があるものを新しいデータセットにする

③ それぞれのモデル ρ を新しいデータで学習

PET with Multiple Masks

- ちょっとよくわからなかったので割愛しますね（便利であるのは間違いない）

GPT-3とくらべます

- それぞれのタスクに対し、32 example
- 20,000 un-labeled example

- GPT-3は（fine-tuningしないので）exampleのみ使う。
- PETは32 exampleでLMのfine-tuningを行い、20,000 dataset に対してSoft-labeling して作ったデータでALBERTを学習する（と思われる）

GPT-3とくらべます

	Model	Params (M)	BoolQ Acc.	CB Acc. / F1	COPA Acc.	RTE Acc.	WiC Acc.	WSC Acc.	MultiRC EM / F1a	ReCoRD Acc. / F1	Avg –
dev	GPT-3 Small	125	43.1	42.9 / 26.1	67.0	52.3	49.8	58.7	6.1 / 45.0	69.8 / 70.7	50.1
	GPT-3 Med	350	60.6	58.9 / 40.4	64.0	48.4	55.0	60.6	11.8 / 55.9	77.2 / 77.9	56.2
	GPT-3 Large	760	62.0	53.6 / 32.6	72.0	46.9	53.0	54.8	16.8 / 64.2	81.3 / 82.1	56.8
	GPT-3 XL	1,300	64.1	69.6 / 48.3	77.0	50.9	53.0	49.0	20.8 / 65.4	83.1 / 84.0	60.0
	GPT-3 2.7B	2,700	70.3	67.9 / 45.7	83.0	56.3	51.6	62.5	24.7 / 69.5	86.6 / 87.5	64.3
	GPT-3 6.7B	6,700	70.0	60.7 / 44.6	83.0	49.5	53.1	67.3	23.8 / 66.4	87.9 / 88.8	63.6
	GPT-3 13B	13,000	70.2	66.1 / 46.0	86.0	60.6	51.1	75.0	25.0 / 69.3	88.9 / 89.8	66.9
	GPT-3	175,000	77.5	82.1 / 57.2	92.0	72.9	**55.3**	75.0	32.5 / 74.8	**89.0 / 90.1**	73.2
	PET	223	79.4	85.1 / 59.4	**95.0**	69.8	52.4	**80.1**	**37.9 / 77.3**	86.0 / 86.5	74.1
	iPET	223	**80.6**	**92.9 / 92.4**	**95.0**	**74.0**	52.2	**80.1**	33.0 / 74.0	86.0 / 86.5	**76.8**
test	GPT-3	175,000	76.4	75.6 / 52.0	**92.0**	69.0	49.4	80.1	30.5 / 75.4	**90.2 / 91.1**	71.8
	PET	223	79.1	87.2 / 60.2	90.8	67.2	**50.7**	**88.4**	**36.4 / 76.6**	85.4 / 85.9	74.0
	iPET	223	**81.2**	**88.8 / 79.9**	90.8	**70.8**	49.3	**88.4**	31.7 / 74.1	85.4 / 85.9	**75.4**
	SotA	11,000	*91.2*	*93.9 / 96.8*	*94.8*	*92.5*	*76.9*	*93.8*	*88.1 / 63.3*	*94.1 / 93.4*	*89.3*

GPT-3とくらべます

- GPT-3と同じくらいのスコア
- パラメータはGPT-3に比べて約1/1000(すごい！)
- SOTAモデルには届いていない
- WiC(単語wが文章S1, S2で同じ意味で使われているか)は苦手

分析

- ρ と$\boldsymbol{v}$ のパターンが精度に大きく影響する。

- 他にもいろいろあるが、よくわからないので割愛

まとめ

- 1/1000のパラメータ数でGPT-3を超えるモデル：ALBERT with PET/iPET を提案した。

- SuperGLUEで勝負するのは若干得意なとこで勝負してる感もある。（GPT-3は翻訳とか文章生成など幅広いタスクに対応してるし…）

- 割愛したPET with Multiple Task がすごそう。