最先端NLP 2024

“What Do Language Models Learn in Context?” (ACL 2024)
“A Theory for Emergence of Complex Skills in Language Models” (arXiv 2023)

持橋大地
統計数理研究所/国立国語研究所
daichi@ism.ac.jp
2024-8-26(月)

国立国語研究所
次世代言語科学研究センター

1本目 (ACL 2024)

What Do Language Models Learn in Context? The Structured Task Hypothesis.

Jiaoda Li* **Yifan Hou*** **Mrinmaya Sachan** **Ryan Cotterell**

{jiaoda.li, yifan.hou, mrinmaya.sachan, ryan.cotterell}@inf.ethz.ch

ETH zürich

- 選んだ理由: Ryan Cotterell祭り(嘘です)

  → 投票が多かったため：大規模言語モデルについて一番大きな謎は、なぜ新しいタスクが解けるのかということ
- この論文の結論: LLMは、
  - 既存のタスクを選んだり、その場で最適化を動かすのではなく
  - **事前学習した関数を組み合わせて問題を解いている**

LLMとIn-Context Learning (ICL)

- プロンプトを与えてその場でタスクを解かせる In-Context Learning (ICL) がなぜできるのかは、まだ理論的に解明されていない
- 3つの大きな説明：
  - 仮説1 LLMは事前学習したタスクの中から、プロンプトに応じたタスクを選んで実行している (Min+ 2022, Xie+ 2022, Wang+ 2023, Wies+ 2023)
  - 仮説2 LLMはGradient Descentのような一般的な機械学習アルゴリズムを学んでおり、ICLは暗黙的にこうした機械学習を実行している (Oswald+ 2023, Akyurek+ 2023, Dai+ 2023)
  - 仮説3 LLMは事前学習で獲得したタスク関数を組み合わせて問題を解いている (Hahn and Goyal 2023)

仮説の検証

- 仮説1,2,3のどれが正しいのかを検証したい
  → プロンプト、あるいは答えを変えて調べる
- 以下では、ICLのプロンプトと答えを
  <p,r> (prompt & response) のペアとして扱う

仮説1: 検証

- 仮説1: LLMはプロンプトからタスクを選んでいるだけ (数学的には、混合モデル)
- 事前学習で遭遇したはずがない<p,r>のペアを与えて予測させる
  - 仮説1が正しければ、これは解けないはず
  - rを一定の関数gでg(r)に変えて、LLMに与える
- 文書分類の3種類のデータセットを使い、L個の<p,g(r)>のDemonstrationを与えてICLを行い、新しいpに対する結果を予測する
  - LLMはLLaMa2-{7B,13B,70B}

仮説1: 結果

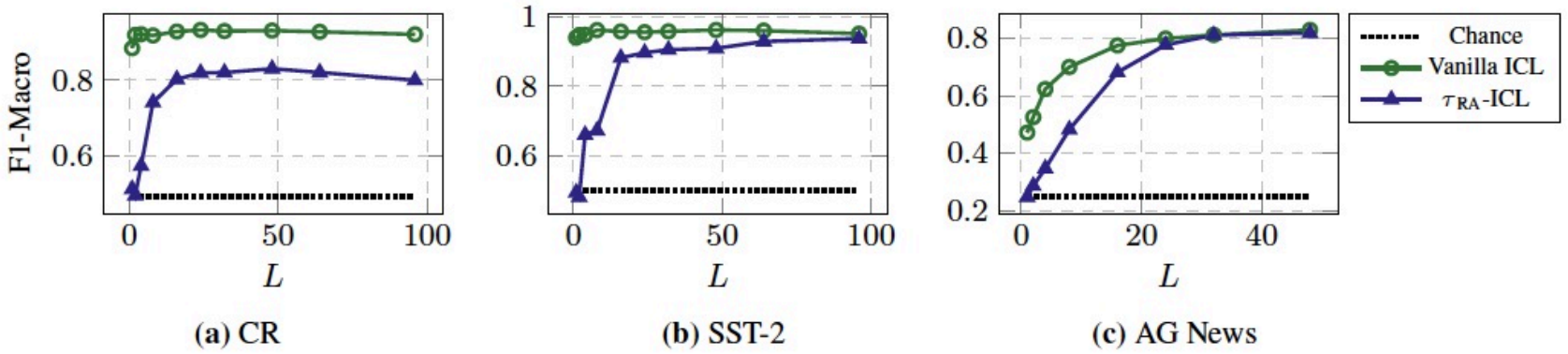


Figure 3: Performance of vanilla ICL and τ_{RA}-ICL on the 3 datasets with different demonstration lengths L. LLaMA2-70B is used. The LLM is able to learn RA tasks as L grows.

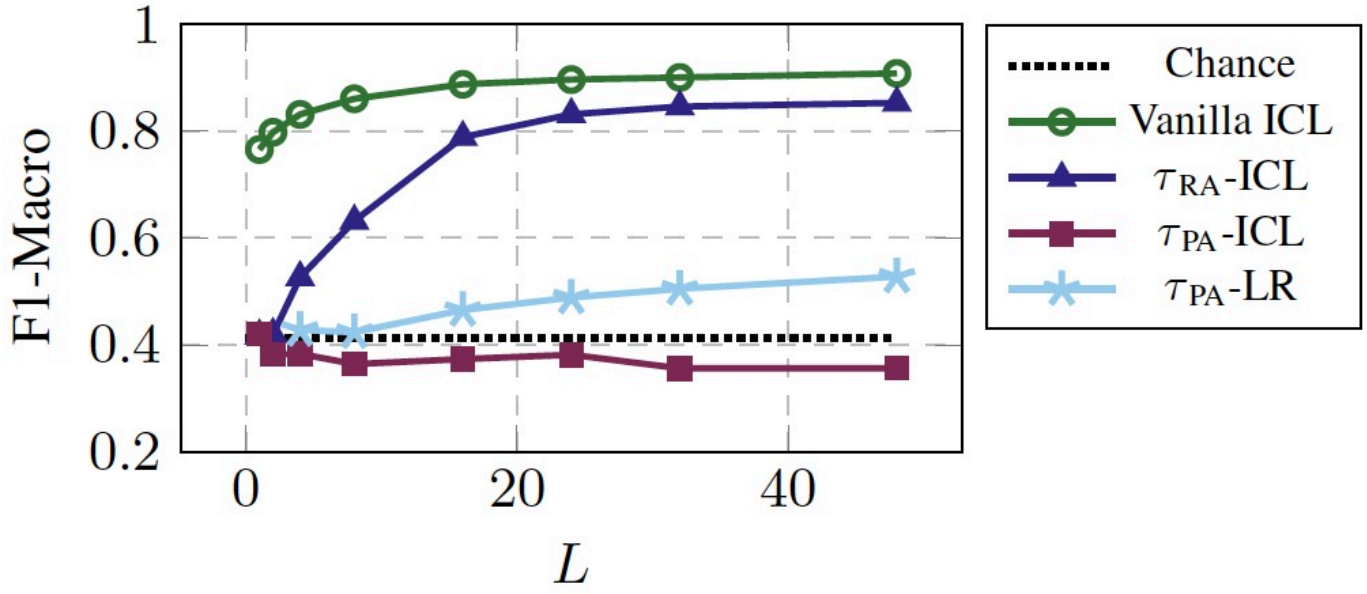
- 結果：これでも解ける　(RA: response-altered)
- 横軸はDemonstrationの数L、縦軸は正解率
- Lが10〜20くらいで性能が上がり、最終的には応答を変えない<p,r>のICLとほぼ同じ性能
  → 仮説1は誤り

仮説2の検証

- 仮説2: ICLは一般的な機械学習を暗黙に行っている
- これが正しければ、<p,r>のうちpを変えても予測できるはず
  - pをある関数hを使って変えた <h(p),r> を見せてICL
  - ICLは勾配法と同じという研究 (Dai+ 2023など) があるため、
    - pからbag of wordsを作ってロジスティック回帰
    - <p,r>をどちらも単語とし、単語埋め込み行列を使って線形回帰

    と比較する
  - 見せる例はL個 (<= Transformerのレイヤー数)

仮説2: 結果 (ロジスティック回帰)



- <p,r>のpを変える (prompt altered=PA) と、まったく学習できない
  - ロジスティック回帰は、Lを増やすと改善する

→ ICLは、ロジスティック回帰を行っているわけではない

仮説2: 結果 (線形回帰)

Dataset	τ_g-Linear (F1-Macro %)	τ_g-ICL (F1-Macro %)	γ_p	γ_s
CR / SST-2	100.0 ± 0.0	92.7 ± 15.2	N.A.	N.A.
AG News	100.0 ± 0.0	93.8 ± 10.8	N.A.	N.A.
DBPedia	99.9 ± 0.8	59.8 ± 19.7	-0.02 (0.59)	0.01 (0.83)

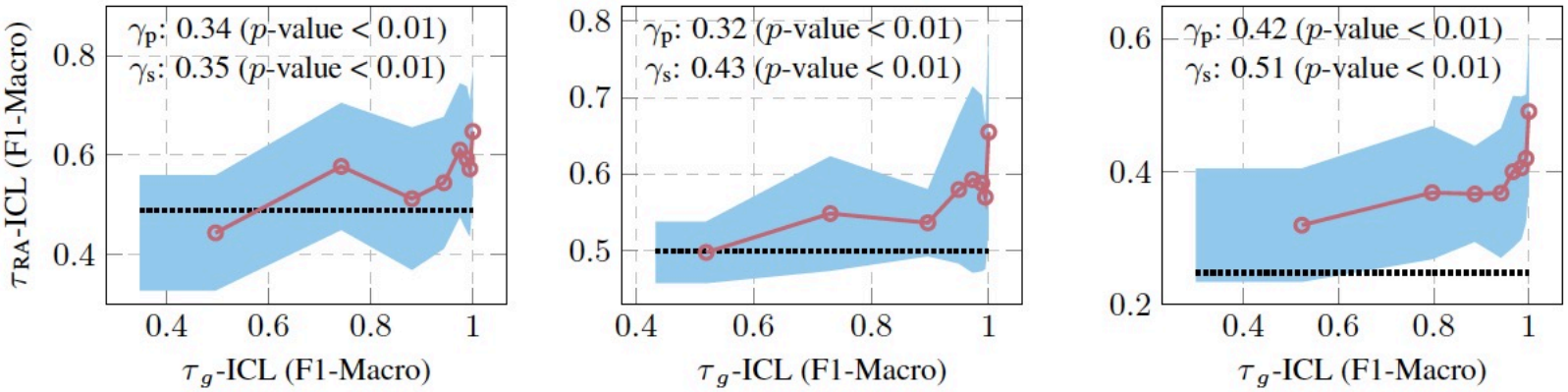
- 線形回帰との比較: 線形回帰は(もちろん)100%正答できる
- ICLは100%ではなく、DBPediaでは6割程度
  かつ、非常に分散が大きい
  - 線形回帰との相関も、ほとんど0

→ ICLは、線形回帰を行っているわけではない

仮説3の検証

- 仮説3: ICLは、事前学習で得られたタスクを組み合わせて新しい問題を解いている
  - 仮説が正しければ、反応を変えたRAタスクの場合、タスクの合成 $\tau_{\mathrm{RA}} = \tau_g \circ \tau$ について τ_g の性能と全体の性能が比例するはず
- 実験
  - 実験1: 色々な τ_g について、その性能で8個のビンに分け、タスク全体の性能との相関をプロット
  - 実験2: より現実的な設定として、τ_g を (1) ランダムな単語 (2) 反意語 (3) 同意語 (4) その分野のキーワード に変換した場合の性能を測定

仮説3: 実験1の結果



- τ_g の性能とICL全体の性能には、弱い相関がみられた
  - 相関係数0.3〜0.5程度で、検定をしても有意な相関

仮説3: 実験2の結果

Dataset	Mapping	F1-Macro (%)	t-test	
			t-value	p-value
CR / SST-2	random	93.2 ± 13.6	N.A.	N.A.
	antonym	97.0 ± 8.5	4.35	< 0.01
	synonym	100.0 ± 0.1	10.71	< 0.01
	keyword	100.0 ± 0.0	10.75	< 0.01
AG News	random	93.8 ± 10.8	N.A.	N.A.
	antonym	99.9 ± 0.3	12.59	< 0.01
	synonym	100.0 ± 0.0	12.72	< 0.01
	keyword	100.0 ± 0.0	12.73	< 0.01
DBPedia	random	59.8 ± 19.7	N.A.	N.A.
	antonym	84.5 ± 20.5	19.42	< 0.01
	synonym	95.8 ± 1.7	40.67	< 0.01
	keyword	93.3 ± 4.9	36.90	< 0.01

- ICLは反意語・同意語・キーワードへの変換を正しく学習できたが、ランダムな変換では性能が落ちた

→ 仮説3を支持

仮説3: 実験3

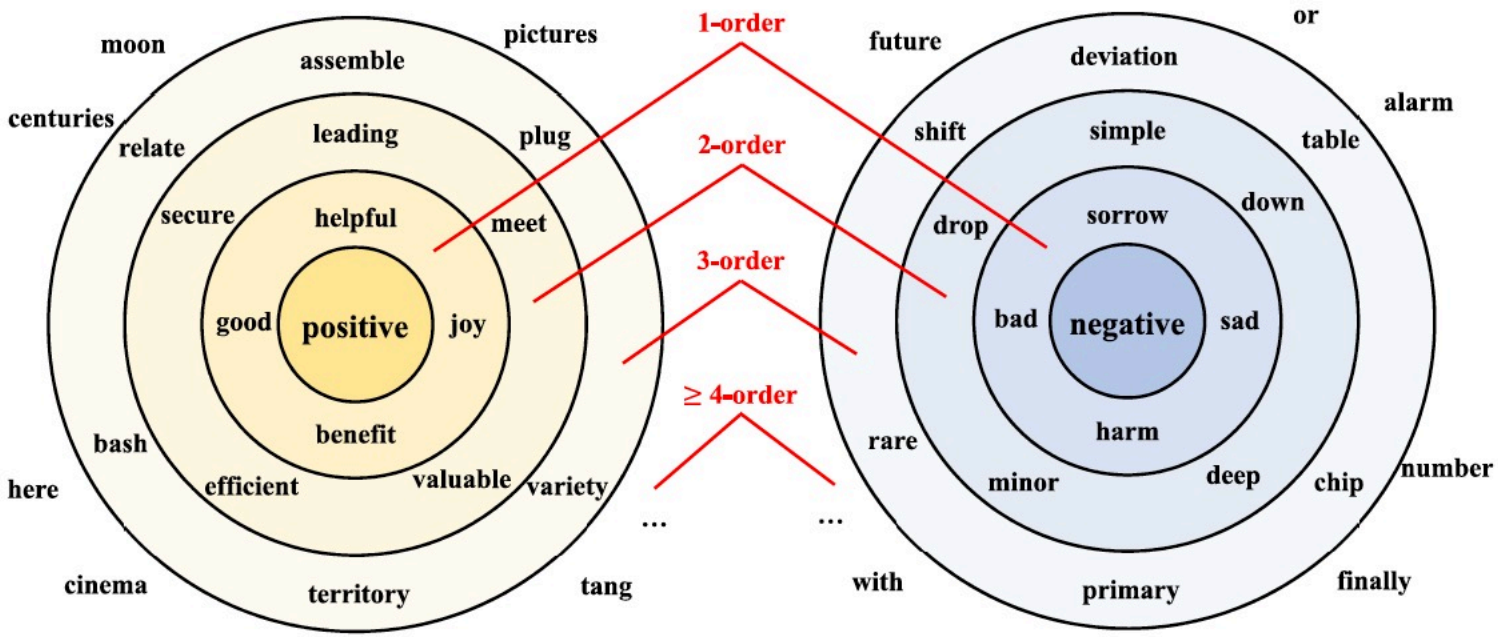
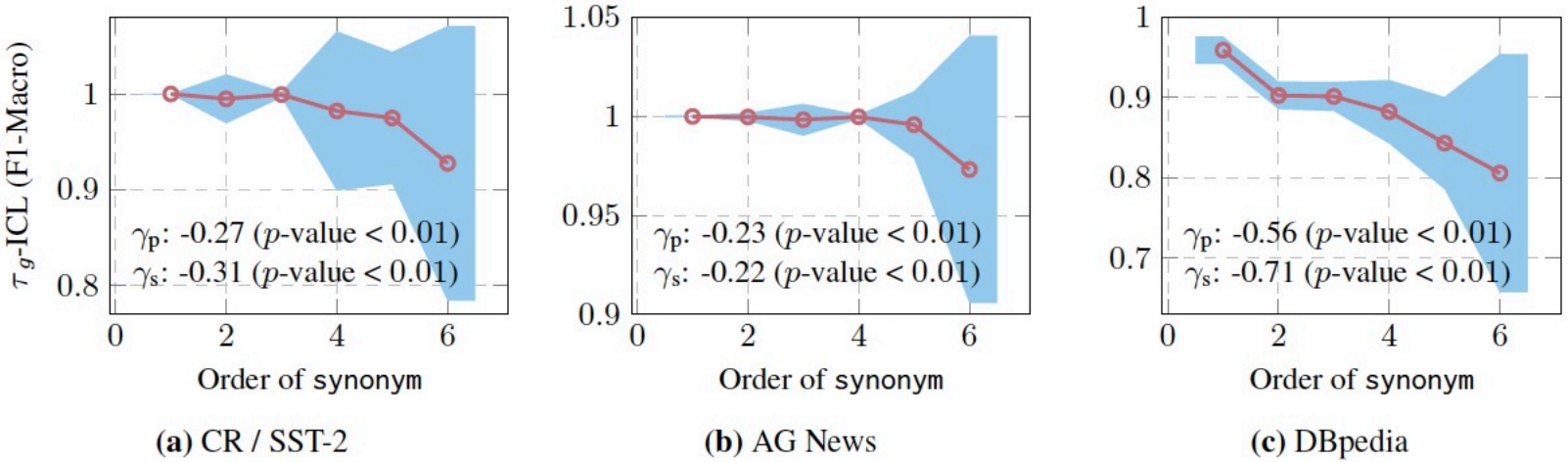


Figure 7: Synonyms of "positive" and "negative". The words between concentric circles represent elements of the candidate sets of high-order synonyms. As the order gets higher, the synonyms become less related to the seed word.

- 同意語への変換を連鎖していく
  - negative→sad→drop→shift→future ; <negative→future>
- 連鎖の深さと性能の相関を調査

仮説3: 実験3の結果



- この複雑なタスクも、非常によく学習できる
  (縦軸の数値に注意)
- 連鎖が深くなるほど、若干性能が落ちる

→ ICLはタスクの合成を行っているという仮説3を支持

ChatGPTの例

これから、回答は日本語で結果だけを伝えてください。

メモリを更新しました

了解しました。

(negate "私は本を読む")

「私は本を読まない」

(お嬢様化 (negate "私は本を読む"))

「わたくしは本を読みませんわ」

(お嬢様化 (cheer-up (negate "私は本を読む")))

「わたくしは本を読みませんけれど、きっと他に楽しいことがございますわ！」

2本目 (arXiv 2023)

A Theory for Emergence of Complex Skills in Language Models

Sanjeev Arora[1, 2], Anirudh Goyal[2]

- LLMの非常に重要な問題: なぜ新しいタスクが解けるのか? に直接数学的に切り込む論文
- 非常に難しい…天才Arora先生のメモ書きのようなもの
- 私にわかる範囲で紹介します

Chinchilla law (Hofmann 2022)

- 学習コーパスのパラメータ数をN, コーパスのサイズをDとおくと、LLMのロス L(N,D) は大略、次の式に従って下がる

$$L(N, D) = A + \frac{B}{N^{0.34}} + \frac{C}{D^{0.28}}$$

余剰クロスエントロピー

- 言語の真の確率分布をp、言語モデルの確率分布をq としたとき、テストデータのクロスエントロピー

$$H(p,q) = -\sum_{x \in \mathcal{L}} p(x) \log q(x)$$

は、情報理論から以下をみたす (i.e. 下限が存在)

$$H(p,q) \geq H(p)$$

- よって重要なのは左辺と右辺の差 $H(p,q) - H(p)$ であり、これを余剰クロスエントロピーと呼んでいる (＝pとqのKLダイバージェンス)
  - Chinchilla lawでは、右辺は定数Aにあたる

Chinchilla law (2)

$$L(N, D) = A + \frac{B}{N^{0.34}} + \frac{C}{D^{0.28}}$$

- Chinchilla lawにおいて、Aは言語の真のエントロピーに当たるので、第2項・第3項が重要
- データが10倍になると、余剰クロスエントロピーは $10^{0.28} = 1.90 \simeq 2$ 分の1になる!
  → 大きく下がる
  - コーパスにS個の潜在的なスキルがあるとき、もしスキル別にコーパスが分解されるなら、この因子は $S^{0.28}/D^{0.28}$ で、ずっと小さい
  - 実際には複数のスキルをまとめて学習することで、大きなゲインがある

潜在スキルとテキスト

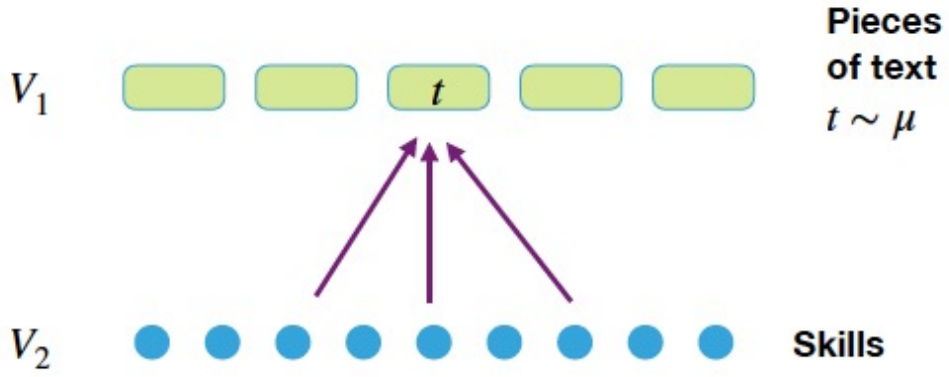
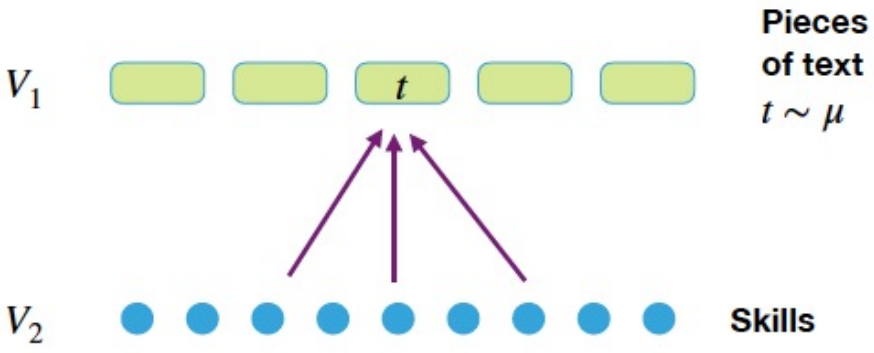


Figure 1: Skill Graph.

- 各テキストの背後に、それを生成した幾つかのスキルがあり、コーパス全体ではS個(〜10^4個など)のスキルがあるとする
  - テキストとスキルの関係は、二分グラフで表せる
- 各テキストは、簡単のためk個のスキルから生成されたと仮定

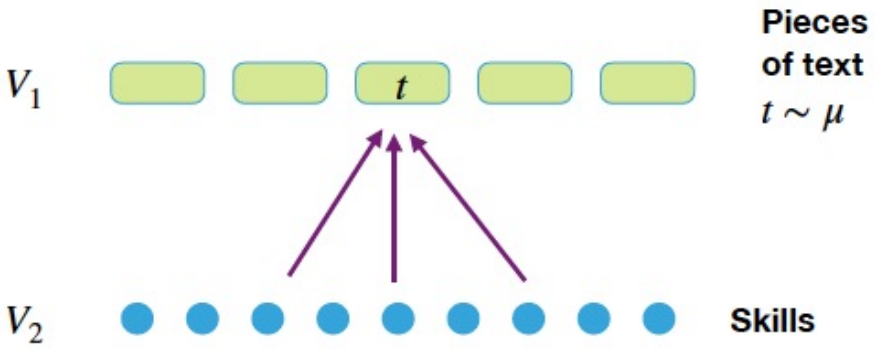
潜在スキルとテキスト(2)

- 潜在スキルの例:
  “The city councilmen refused the demonstrators a permit because they feared violence.”
  の理解には、
  - “they”の参照の理解
  - “councilmen”, “permit”, “demonstrators”などに関する世界知識
  - becauseで表される因果関係の理解　などが必要



潜在スキルとテキスト (補足)

- 論文で積極的には使っていないが、これは
  心理統計学での多次元項目反応理論、または認知診断
  モデルと非常に似ている
- 生成モデル:
  - $\mathbf{s} \sim p(\mathbf{s})$ で潜在スキルベクトルsを生成
  - $t \sim p(t|\mathbf{s})$ でsから観測されたテキストを生成 (GEN)
  - (tに従って質問に答える) (CLOZE)

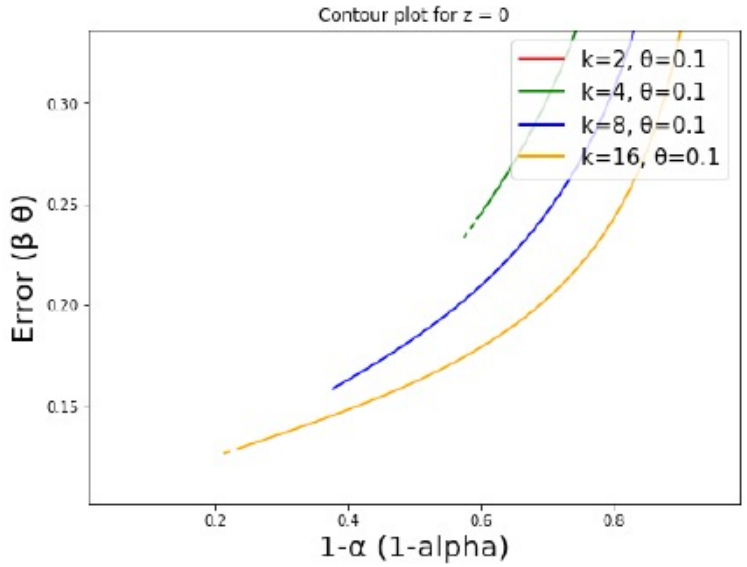


ランダムグラフ理論による結果

- コーパスのうち、CLOZE質問に誤って答えるテキストの割合をθとすると、次式が高確率で成り立つ

$$H(\theta) + k\theta(H(\beta\alpha) - \beta\alpha \log \frac{1}{\alpha} - (1 - \beta\alpha) \log(\frac{1}{1-\alpha})) < 0$$

  - H()は二値エントロピー関数
- これをプロットすると、



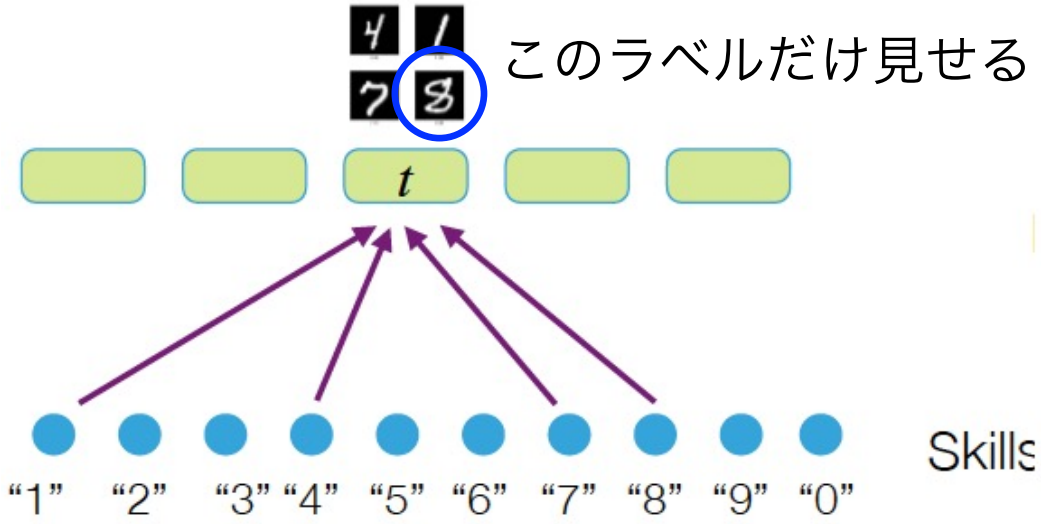
・1テキスト当たりのスキルの数kが増えるほど、誤り率θは下がる
・上記の性質は、数学的に証明できる
→ テキストが難しいほど、言語モデルの能力が向上する

Paucity of stimulus

- GPT-2レベルの10^{10}個のトークンで、10^4個の各スキルが学習されていたとする
- 定理によると、10^{13}個のトークン(現在の多くのLLM)では、2^3=8個のスキルを用いた複合タスクが解ける
  - 全部を見るには、$(10^4)^8=10^{32}$個のトークンが必要なはず

→ Paucity of Stimulus (刺激の不足) 問題に対する解答

実験



- MNISTから、ランダムな4つの数字を2x2に配置した画像を1000個作成
  - 画像のラベルは、4つの数字のうちランダムな1つだけ
- 学習済みViT-CLIPをこの画像で訓練して、テストデータの各画像に含まれる「4つの数字」を予測するタスク
  → 92%の場合で成功!
  - 10000個ではなく1000個の画像で、数字は各1個しか見せていないのに、学習が汎化している

まとめ

- LLMについて一番の謎は、「なぜ教えていないタスクが解けるのか?」ということ
- 1番目の論文は実験的にこれを検証して、「事前学習されたタスクの組み合わせ」で問題を解いていることを示唆
- 2番目の論文は理論的に、ランダムな二分グラフからテキストが生まれたと考えることで、スキルの組み合わせを全部見なくても各スキルが学習できることを示唆
  - Transformerの具体的な動作などは使っていない“熱力学的な議論”なので、理論的・実験的に詳しく調べる余地は大いにある