Meta-Learning to Compositionally Generalize

Henry Conklin, Bailin Wang, Kenny Smith, Ivan Titov

読む人：吉川将司 (東北大), 2021/09/17, 第13回最先端NLP勉強会

概要：メタ学習で構成性に強いNLPモデル

- 構成性：小さな単位の部品 (単語, 句) の使い回して理解する能力
  - NNモデルは長い単位で丸覚えしがち (Hupkes et al., 2020, Keysers et al., 2020)
- **メタ学習** (learn-to-learn) と類似度尺度だけで構成性awareなNLPモデル
  - few-shot学習、分野適応、学習アルゴリズムやハイパラの学習など
  - NLPでの位置づけは？ -> 構成性を捉えた学習を可能にできる？
- モデルの構造を工夫しなくてもSCAN, COGSで性能改善

e.g., 標準的なTransformer　　構成性を捉えていれば簡単な意味解析タスク

jump => JUMP
jump left => LTURN JUMP
SCAN (Lake and Baroni, 2018)

Henry liked a cockroach =>
```
like.agent(x_1, Henry)
& like.theme(x_1, x_3)
& cockroach(x_3)
```
COGS (Kim & Linzen, 2018)

構成性

（よく見るもの）the meaning of a whole is a function of the meanings of the parts and of the way they are syntactically combined (Partee 1995)

- 構成性を捉えて構成性テストをパスするには (cf. Hupkes et al., 2020)：
  - systematicity: 既知の要素を新たな方法で組み合わせる
    学習：The **cat** gives the **dog** a gift → 評価：The **dog** gives the **cat** a gift
  - productivity: 学習時に見た系列より長いものへの汎化
    学習：The cat gives the dog a gift → 評価：The cat gives the dog a gift **and the bird a gift**
  - primitive application: 学習時に単独で出現する語を組み合わせて使える
    学習：**made** → 評価：The cat **made** the dog a gift
- (余談) 弱構成性 (Szabó 2008, Dankers et al., 2021)

COGSはラムダ式でprimitiveを表すので項構造情報から学習することに相当

$\lambda a.\lambda b.\lambda e.\texttt{draw.agent}(e, b)\ \&\ \texttt{draw.theme}(e, a)$

- *The meaning of a complex expression is determined by* ***the meaning of*** ***its structure*** *and the meanings of its constituents.*
  - “meaning”は文脈に依存できる -> こちらの方がNLP的に扱いやすいかも？

構成性と現在のNLPアプローチ

1. 構成性を捉えた汎化を目指すことはi.i.d.の仮定は相性が悪そう
   学習時にrareなイベントでも構成的なものであれば処理できないといけない
2. in-domainタスクに対しても解く作戦が色々あり得てしまう

   - 人間的な文処理でなくていい、superficial cueが典型的な話
     (Gururangan et al., NAACL2018など)
   - NNは大きな単位で丸暗記しがち？だが小さな構成素と例外の記憶が重要
     (Hupkes et al., 2020, Keysers et al., 2020)

- Connectionismのアプローチで構成性は扱えるか？
  否定派 (Fodor and Pylyshyn, 1988)
  - DNNの研究で回答するには、上の問題をよく考える必要性がありそう
  - 本研究はメタ学習 (MAML; Fin et al., 2017) でアプローチする

MAML for few-shot learning (Fin et al., 2017)

learn to learn by meta-learning

```
def MAML_train():
1: θ ← (random init)
2: for 1 to meta_epoch:
3:   D_train, D_dev ~ p(task)
4:   θ' ← (init as θ)   // 🐶🐱分類タスクなど
5:   for 1 to epoch:
6:     θ' ← update(θ', D_train)
7:   L_dev ← loss(θ', D_dev)
8:   θ ← θ - α∇_θ L_dev
```

- θ が良質なら小さい D_{train} で低い損失達成
- θ からL_{dev}は微分可能な形で繋がってる
  - θ をいかにずらせばより良く学習できるか ($=\nabla_\theta L_{\text{dev}}$) が計算可能
- 最終的な θ は次のfew-shot taskで有用

「見る」ことができれば新しい物体カテゴリも簡単に識別出来るようになるはず！

提案法：MAMLで構成性を捉えるには？

```
def MAML_train():
1: θ <- (random init)
2: for 1 to meta_epoch:
3:    D_train, D_dev ~ p(task)
4:    θ' <- (init as θ)
5:    for 1 to epoch:
6:       θ' <- update(θ', D_train)
7:    L_dev <- loss(θ', D_dev)
8:    θ <- θ - α∇_θ L_dev
```

D_{train}:
The **cat** **gives** the dog a gift.
Henry **made** a cake.
…

D_{dev}:
The dog **gives** the **cat** a gift.
The dog **gives** a gift to the **cat**.
The **cat** **made** the dog a gift.
…

- D_{train} と D_{dev} で語彙を共有しつつ構造が異なっていればどうか？
  - D_{train} は語彙知識を提供可
- θ はいかに語を組み合わせて全体の意味を構築するか (≒文法) を捉える必要あり

類似度尺度に基づくサンプリング

- **構成性を考慮**：語彙を共有しつつ異なった構文の事例がほしい
- まず $D_{\texttt{train}}$をランダムに抽出 -> $D_{\texttt{dev}}$を類似度ベースにサンプル
  - i.e., $p(x, y \mid x', y') \propto \exp(\kappa([x, y], [x', y'])/\tau)$
    x: 文, y: 意味表現
- κ: 負のLevenshtein距離、文字列、木カーネル
  共通の部分単語列 (木) がいくつあるか
  - 木カーネルは論理式 (木) を扱うCOGSでのみ使用
    構文解析して木を作ってもいい？
- 全学習事例からのサンプリングは重いため近似
  - $\tilde{p}(x, y \mid x', y') = \lambda p_{1000}(x, y \mid x', y') + (1 - \lambda)(1/N)$
    類似度上位1000個だけキャッシュ　　他はuniform

Source Example: The girl changed a sandwich beside the table .

Neighbours using Tree Kernel	Similarity
A sandwich changed .	0.55
The girl changed .	0.55
The block was changed by the girl .	0.39
The girl changed the cake .	0.39
change	0.32
Neighbours using String Kernel	
The girl rolled a drink beside the table .	0.35
The girl liked a dealer beside the table .	0.35
The girl cleaned a teacher beside the table .	0.35
The girl froze a bear beside the table .	0.35
The girl grew a pencil beside the table .	0.35
Neighbours using LevDistance	
The girl rolled a drink beside the table .	-2.00
The girl liked a dealer beside the table .	-2.00
The girl cleaned a teacher beside the table .	-2.00
The girl froze a bear beside the table .	-2.00
The girl grew a pencil beside the table .	-2.00

本研究で採用されてる近似 & 手法のまとめ

```
def MAML_train():
1: θ <- (random init)
2: for 1 to meta_epoch:
3:     D_train, D_dev ~ p(task)                    ①
4:     θ' <- (init as θ)
5:     # only 1 step update                        ②
6:     θ' <- update(θ', D_train)
7:     L <- loss(θ, D_train) + loss(θ', D_dev)
8:     θ <- θ - α∇_θ L                             ③
```

① D_{train}とD_{dev}：128事例1バッチ

② 実際は内側のループは1 stepのみ

- 巨大な計算グラフを避けるため

③ D_{train}上の損失も影響させる

- この近似の基での解釈：
  - D_{train}とD_{dev}の勾配の内積を最大化？

$$
\begin{aligned}
\mathcal{L}_\tau(\theta) &= \mathcal{L}_{\mathcal{B}_s}(\theta) + \mathcal{L}_{\mathcal{B}_t}(\theta') \\
&= \mathcal{L}_{\mathcal{B}_s}(\theta) + \mathcal{L}_{\mathcal{B}_t}(\theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{B}_s}(\theta)) \\
&\approx \mathcal{L}_{\mathcal{B}_s}(\theta) + \mathcal{L}_{\mathcal{B}_t}(\theta) - \\
&\quad \alpha(\nabla_\theta \mathcal{L}_{\mathcal{B}_s}(\theta) \cdot \nabla_\theta \mathcal{L}_{\mathcal{B}_t}(\theta))
\end{aligned}
$$

(Wang et al., 2018より)
逆にこれで構成性的な汎化が出来るのかよくわからない？

評価実験①：SCAN (Lake and Baroni, 2018)

構成性を捉えてれば簡単に解けるはずの人工的なタスク

1-3の性質の違いは不明
(構築時のseedの違い)

Model	MCD1	MCD2	MCD3
LSTM	27.4 ±8.2	31.0 ±0.4	9.6 ±3.7
w. Uni-MAML	44.8 ±5.4	31.9 ±3.4	10.0 ±1.4
w. Lev-MAML	**47.6** **±2.3**	**35.2** **±3.9**	11.4 ±3.0
w. Str-MAML	42.2 ±2.6	33.6 ±4.3	11.4 ±2.2
Transformer	2.6 ±0.8	3.1 ±1.0	2.3 ±1.3
w. Uni-MAML	2.8 ±0.7	3.2 ±1.0	3.2 ±1.6
w. Lev-MAML	4.7 ±1.8	6.7 ±1.4	6.5 ±1.2
w. Str-MAML	2.8 ±0.6	5.6 ±1.6	6.7 ±1.4

Table. 2より一部略

モデルの構造を工夫すれば100%解ける (Liu et al., 2020)
とかはあるがbasicな構造での向上がポイント

maximum compound divergence

- MCD: 学習/評価が構成性のテストになっているかの指標を最大化して構築
  (Keysers et al., 2020)
- MAMLなし/uniform samplingより有効
  - Levenshtein距離を使うものが有効

Transformerには有効でなかった？

jump	⇒	JUMP
jump left	⇒	LTURN JUMP
jump around right	⇒	RTURN JUMP RTURN JUMP RTURN JUMP RTURN JUMP
turn left twice	⇒	LTURN LTURN
jump thrice	⇒	JUMP JUMP JUMP
jump opposite left and walk thrice	⇒	LTURN LTURN JUMP WALK WALK WALK
jump opposite left after walk around left	⇒	LTURN WALK LTURN WALK LTURN WALK LTURN WALK LTURN LTURN JUMP

評価実験②：COGS (Kim & Linzen, 2020)

学習データと同じ分布　要汎化

Model	Test	Gen
LSTM	99.7	34.5 ±4.5
w. Uni-MAML	99.7	36.4 ±3.6
w. Lev-MAML	99.7	36.4 ±5.2
w. Str-MAML	99.7	36.8 ±3.5
w. Tree-MAML ⋯	99.7	**41.0** ±4.9
Transformer	99.5	58.6 ±3.7
w. Uni-MAML	99.6	64.4 ±4.0
w. Lev-MAML	99.7	64.9 ±6.3
w. Str-MAML	99.6	64.8 ±5.5
w. Tree-MAML	99.6	**66.7** ±4.4

Table. 3より一部略

- 意味解析 (学習/評価=24,155/21,000)
  CFGによる生成でありながら構造のカバレッジを意識
- Linaは学習時は主語としてのみ出現 -> 目的語で現れても扱えるか
  さらに名詞句にPP句がつく亜種も
- 他に文構造の変化 (e.g., 受動能動), 再帰の深さの汎化 (say, believe等)なども

Case	Training	Generalization
S.3.1. Novel Combination of Familiar Primitives and Grammatical Roles		
Subject → Object (common noun)	A **hedgehog** ate the cake.	The baby liked the **hedgehog**.
Subject → Object (proper noun)	**Lina** gave the cake to Olivia.	A hero shortened **Lina**.
Object → Subject (common noun)	Henry liked a **cockroach**.	The **cockroach** ate the bat.
S.3.2. Novel Combination Modified Phrases and Grammatical Roles		
Object modification → Subject modification	Noah ate **the cake on the plate.**	**The cake on the table** burned.

```
like.agent(x_1, Henry)
& like.theme(x_1, x_3)
  & cockroach(x_3)
```

まとめと感想

- **まとめ**：MAMLを使って構成性を捉えた汎化を可能に
  - 要素を使い回さなければ解けないタスクを自動生成

構成性を考慮した分布の外を意識 & 学習の仕方にバイアス

- **感想**：構成性の問題を木を介さず機械学習の言葉でアプローチできて面白い
  - メタ学習のNLPにおける意義？
    - (learn to learnができる) = (新しい語にすぐに適応できる) = 文法能力？
  - 分野適応に言及できる技術ならメタ学習でなくてもいい？

参考文献

- Andrychowicz et al., Learning to learn by gradient descent by gradient descent, NeurIPS2016
- Keysers et al., Measuring Compositional Generalization: A Comprehensive Method on Realistic Data, ICLR2020
- Dankers et al., The paradox of the compositionality of natural language: a neural machine translation case study, arxiv
- Lake and Baroni, Generalization without Systematicity: On the Compositional Skills of Sequence-to-Sequence Recurrent Networks, ICML2018
- Kim and Linzen, COGS: A Compositional Generalization Challenge Based on Semantic Interpretation, EMNLP2020
- Hupkes et al., Compositionality decomposed: how do neural networks generalise?, IJCAI2020
- Finn et al., Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks, ICML2017
- Wang et al., Meta-Learning for Domain Generalization in Semantic Parsing, NAACL2021
- Liu et al., Compositional Generalization by Learning Analytical Expressions, NeurIPS2020
- Stanford Encyclopedia of Philosophy (Compositionality) https://plato.stanford.edu/entries/compositionality/