Pre-trained Language Models Can be Fully Zero-Shot Learners

Xuandong Zhao , Siqi Ouyang , Zhiguo Yu , Ming Wu , Lei Li

ACL 2023

https://aclanthology.org/2023.acl-long.869.pdf

ACL読み会2023

名古屋大学 武田・笹野研究室

M1 大鹿 雅史

概要

- PLMをZero-shotでNLUタスクなどの性能を向上させる手法(NPPrompt)の提案
  - ・テキスト分類、感情分析など
- 入力プロンプトの[MASK]の単語予測とラベル名の近傍の単語を利用
- 他のZero-shot手法を大幅に上回る性能を示す

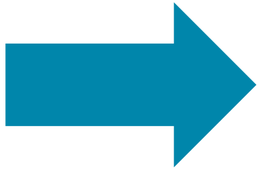
- 選定理由
  - ・手法がシンプルでわかりやすい
  - ・LLMによってZero-shotでの研究は増えてきそう

背景

- PLMの台頭によりNLUタスクの大きな進歩が見られる
  - ・事前学習によってコーパスから知識を獲得する
  - ・タスク特有のデータセットでfine-tuningすることでNLPタスクの性能が向上する
- Fine-tuningの問題点
  - ・ラベル付きデータセットの構築に人手のコスト
  - ・モデルを学習させる計算コスト
  - ・タスクごとにモデルを保存する必要

背景

- PLMの台頭によりNLUタスクの大きな進歩が見られる
  - ・事前学習によってコーパスから知識を獲得する
  - ・タスク特有のデータセットでfine-tuningすることでNLPタスクの性能が向上する
- Fine-tuningの問題点
  - ・ラベル付きデータセットの構築に人手のコスト
  - ・モデルを学習させる計算コスト
  - ・タスクごとにモデルを保存する必要



Zero-shot でタスクが解けるとコストが少なく済む

関連研究

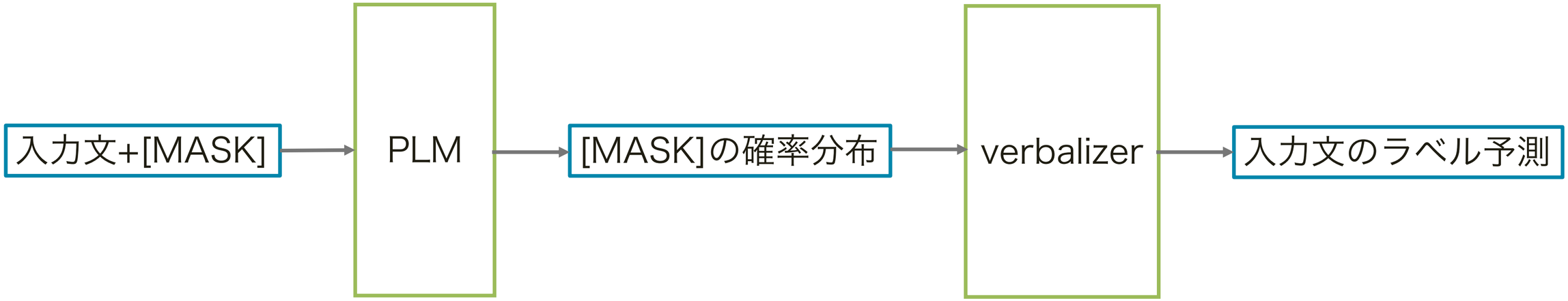
- Prompting
- Verbalizer Design
- Zero-shot Text Classification

関連研究

- Prompting
  - ・GPT-3の成功によってより注目を集めている技術
  - ・さまざまな自然言語処理タスクで性能を示している

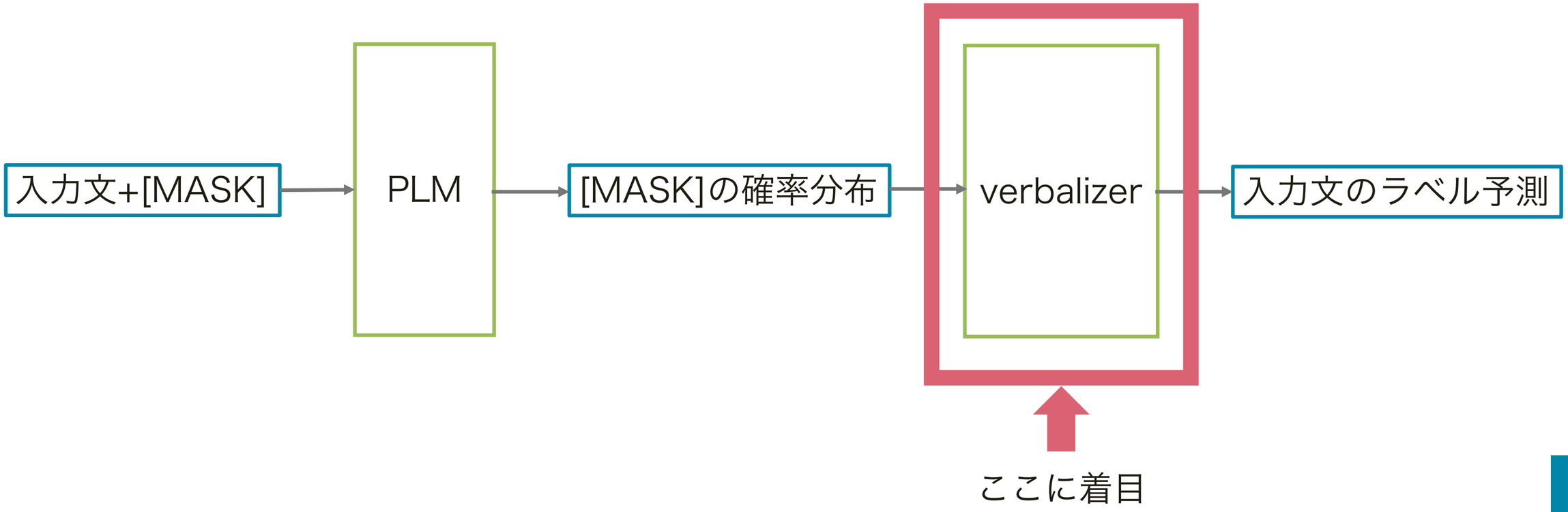
関連研究

- Prompting
  - ・GPT-3の成功によってより注目を集めている技術
  - ・さまざまな自然言語処理タスクで性能を示している



関連研究

- Prompting
  - ・GPT-3の成功によってより注目を集めている技術
  - ・さまざまな自然言語処理タスクで性能を示している



関連研究

Verbalizer Design

・人手で作成 → バイアスが発生

・自動でverbalizerを構築する手法が提案

1. Auto-L

   ラベル名を表す単語の候補集合を元にモデルをfine-tuningしverbalizerを再定義する

2. AutoPrompt

   勾配ベースの探索でラベルを表すverbalizer(トリガートークン)を定義

3. KPT

   外部知識を利用してverbalizerを洗練する

関連研究

Verbalizer Design

・人手で作成 → バイアスが発生

・自動でverbalizerを構築する手法が提案

1. Auto-L

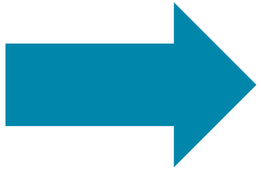
   ラベル名を表す単語の候補集合を元にモデルをfine-tuningしverbalizerを再定義する

2. AutoPrompt

   勾配ベースの探索でラベルを表すverbalizer(トリガートークン)を定義

3. KPT

   外部知識を利用してverbalizerを洗練する



モデルの勾配情報や外部知識が必要

関連研究

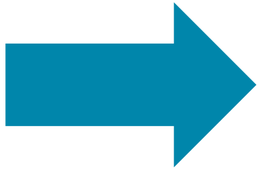
Zero-shot Text Classification

- SimPTC
  - 外部知識を元に各クラスのアンカーを決める
  - アンカーとの類似度を元にクラスタリングを行いテキスト分類
- LOTClass
  - 各クラスを表す単語を複数定義
  - MLMの単語予測と各クラスの単語の一致度からテキスト分類
  - クラス名をもとにMLMの自己学習を行う

関連研究

Zero-shot Text Classification

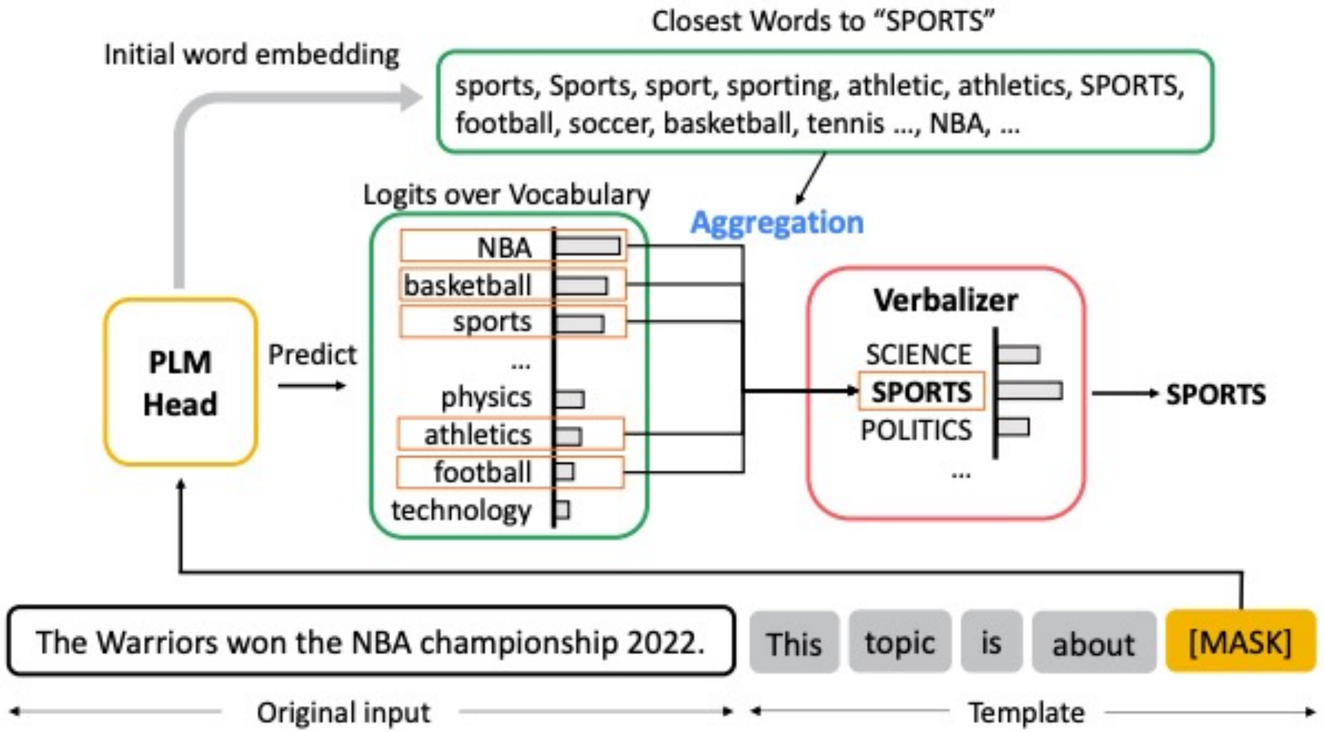
- SimPTC
  - 外部知識を元に各クラスのアンカーを決める
  - アンカーとの類似度を元にクラスタリングを行いテキスト分類
- LOTClass
  - 各クラスを表す単語を複数定義
  - MLMの単語予測と各クラスの単語の一致度からテキスト分類
  - クラス名をもとにMLMの自己学習を行う



外部知識や学習コストがかかる

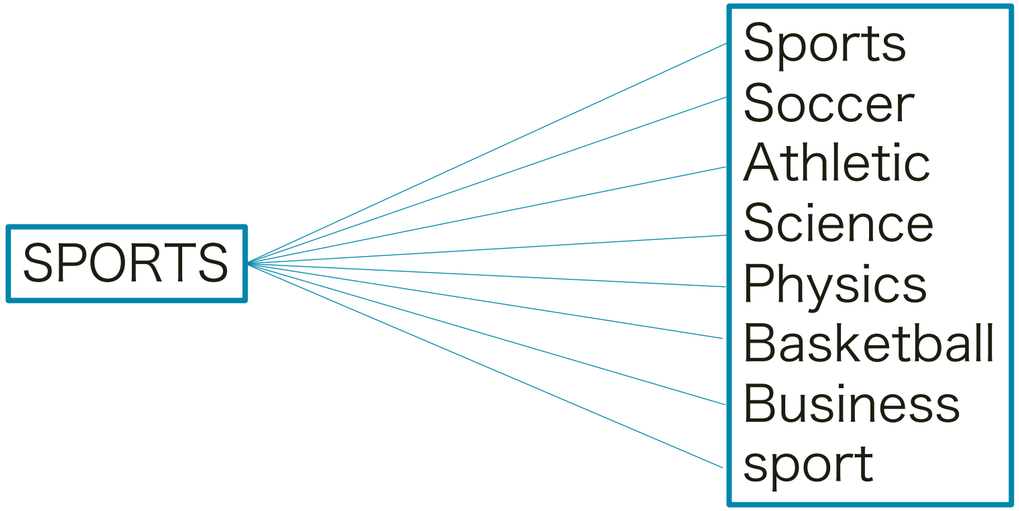
NPPrompt : NonParametric Prompting

- 手法
  1. k-近傍verbalizerの構築
  2. ラベルの予測



NPPrompt : k-近傍verbalizerの構築

- PLMが持つ語彙とラベル名の埋め込みの類似度を計算
- 類似度が高いTop-kの単語からverbalizerを構築する

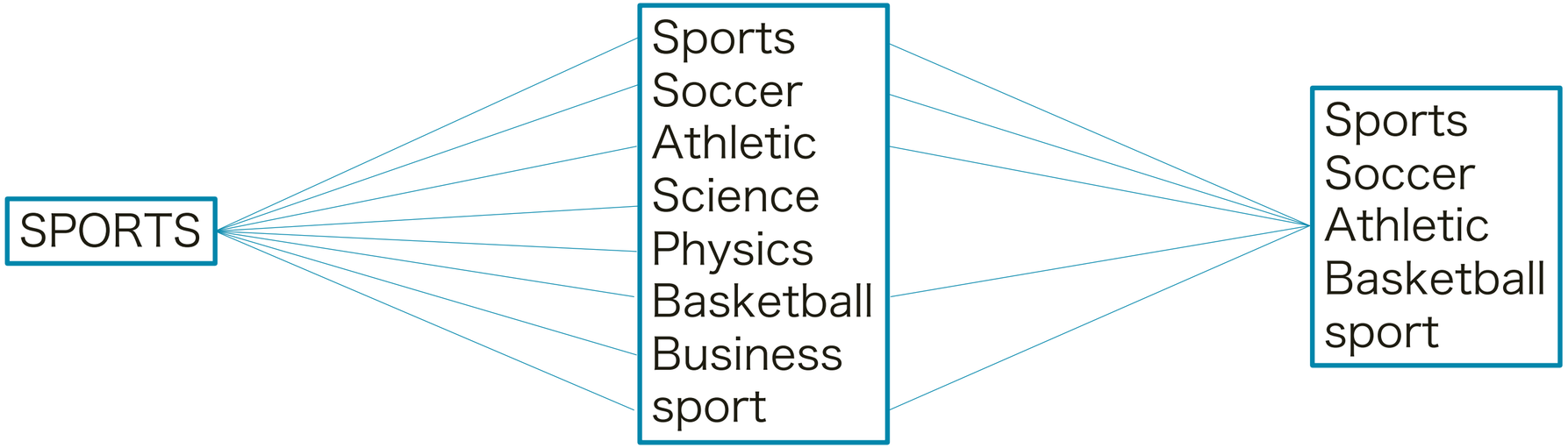


$$S\left(emb(v_i), emb(y_j)\right) = \frac{emb(v_i)}{||emb(v_i)||} \cdot \frac{emb(y_i)}{||emb(y_i)||}$$

v_i:verbalizer内の単語, y_i:ラベル名

NPPrompt : k-近傍verbalizerの構築

- PLMが持つ語彙とラベル名の埋め込みの類似度を計算
- 類似度が高いTop-kの単語からverbalizerを構築する



$$S\left(emb(v_i), emb(y_j)\right) = \frac{emb(v_i)}{||emb(v_i)||} \cdot \frac{emb(y_i)}{||emb(y_i)||} \qquad M(y_j) = \underset{v_i \in V}{\text{Top k}}\{S(emb(v_i), emb(y_j))\}$$

v_i:verbalizer内の単語, y_i:ラベル名

NPPrompt：ラベルの予測

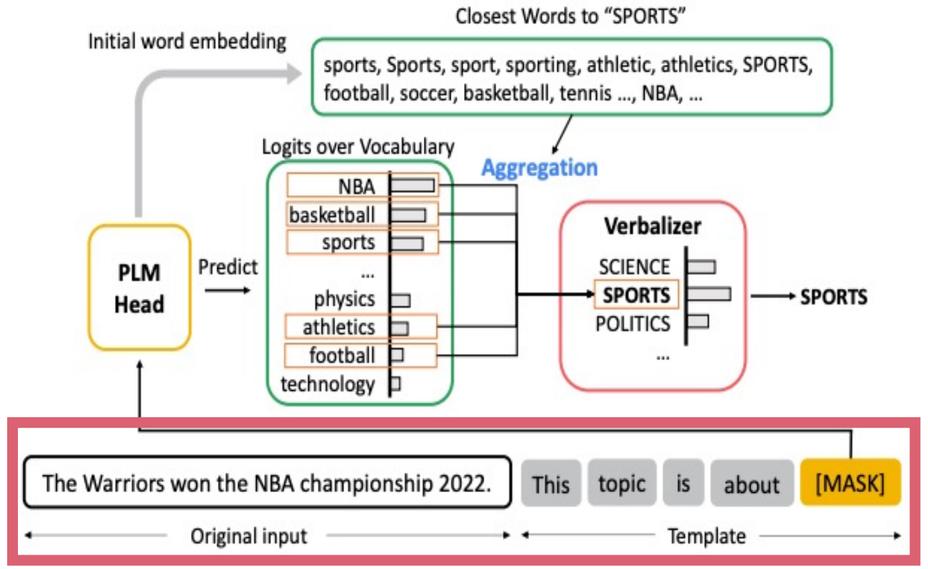
- 分類するテキストを元に[MASK]を利用したプロンプトを作成

  The warriors won the NBA championship 2022.

  →

- プロンプトをPLMに入力し[MASK]の予測Logitを出力

- Verbalizer内の単語からラベルを予測



NPPrompt：ラベルの予測

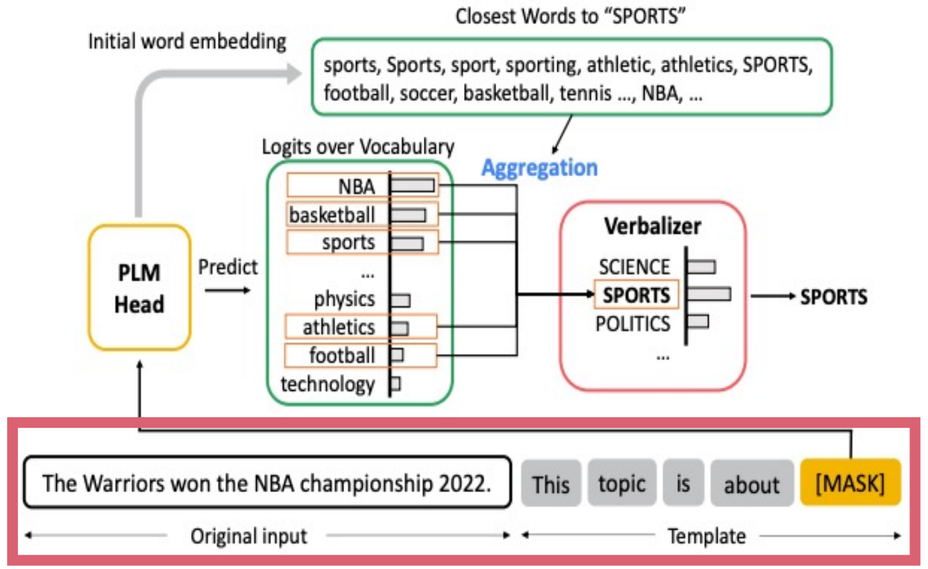
- 分類するテキストを元に[MASK]を利用したプロンプトを作成

  The warriors won the NBA championship 2022.

  → The warriors won the NBA championship 2022. This topic is about [MASK].

- プロンプトをPLMに入力し[MASK]の予測Logitを出力

- Verbalizer内の単語からラベルを予測



NPPrompt：ラベルの予測

- 分類するテキストを元に[MASK]を利用したプロンプトを作成

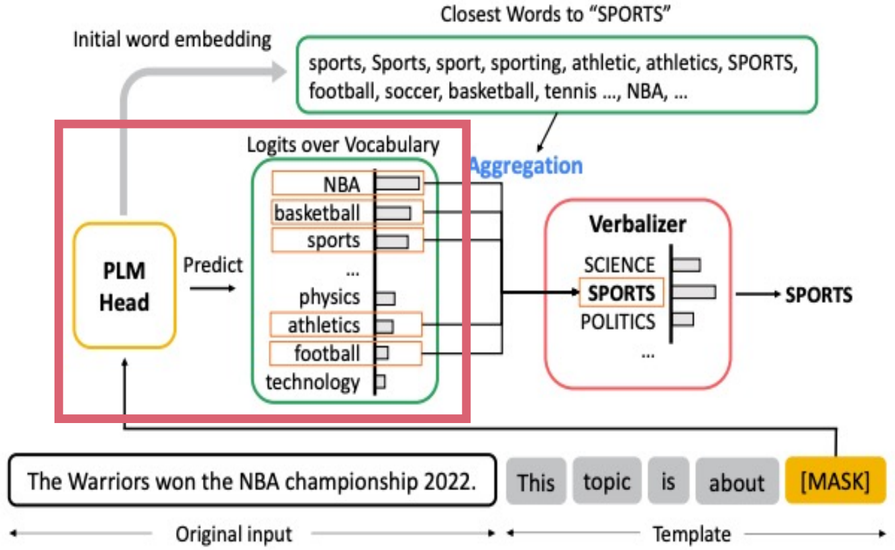
  The warriors won the NBA championship 2022.

  → The warriors won the NBA championship 2022. This topic is about [MASK].

- プロンプトをPLMに入力し[MASK]の予測Logitを出力

$$\Theta\left([MASK] = v_i \middle| x_{prompt} = \tau(x)\right)$$

- Verbalizer内の単語からラベルを予測



NPPrompt：ラベルの予測

- 分類するテキストを元に[MASK]を利用したプロンプトを作成

  The warriors won the NBA championship 2022.

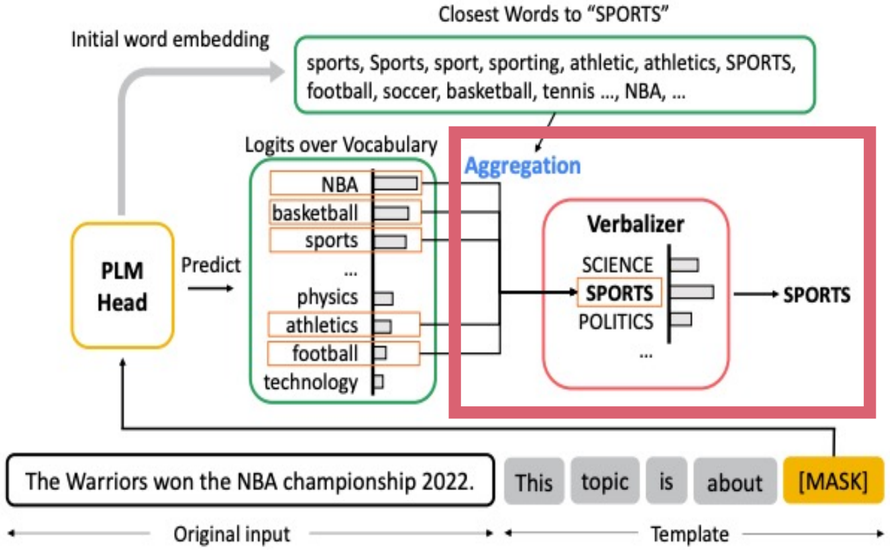
  → The warriors won the NBA championship 2022. This topic is about [MASK].

- プロンプトをPLMに入力し[MASK]の予測Logitを出力

$$\Theta\left([MASK] = v_i \middle| x_{prompt} = \tau(x)\right)$$

- Verbalizer内の単語からラベルを予測

$$Q(y_j|x) = \sum_{v_i \in M(y_j)} w(v_i, y_j) \cdot \Theta([MASK] = v_i | x_{prompt} = \tau(x))$$



NPPrompt：ラベルの予測

- 分類するテキストを元に[MASK]を利用したプロンプトを作成

  The warriors won the NBA championship 2022.

  → The warriors won the NBA championship 2022. This topic is about [MASK].

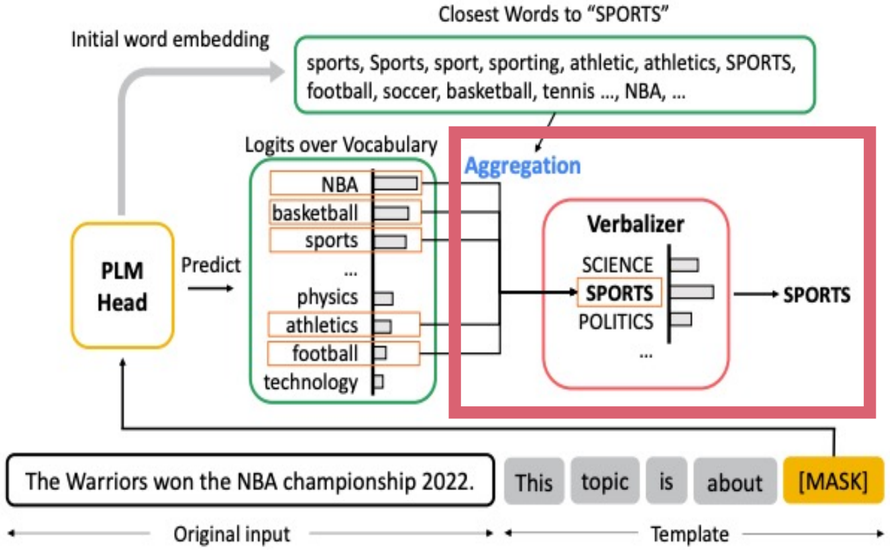
- プロンプトをPLMに入力し[MASK]の予測Logitを出力

$$\Theta\left([MASK] = v_i \middle| x_{prompt} = \tau(x)\right)$$

- Verbalizer内の単語からラベルを予測

$$\mathrm{Q}(y_j|x) = \sum_{v_i \in M(y_j)} w(v_i, y_j) \cdot \Theta([MASK] = v_i | x_{prompt} = \tau(x))$$

ソフトマックス関数



実験設定

- 利用するデータセット

  ① テキスト分類

  ・トピック分類

  AG News：ニュースのトピック分類 (4クラス)

  DBPedia：Wikipediaのトピック分類 (14クラス)

  ・感情分析

  IMDB：映画レビューの感情分析 (positive：negative)

  Amazon：レビューの感情分析 (positive：negative)

  ② GLUEベンチマーク (自然言語推論、含意関係認識など)

- 利用モデル：RoBERTa-large(355M)

実験設定

■ 利用したプロンプトやラベル名の例①

Dataset	Template	Label Names	k
AG News	A [MASK] news : x .	category 1: *world, politics* category 2: *sports* category 3: *business* category 4: *technology, science*	12
DBPedia	x_1 x_2 In this sentence, x_1 is a [MASK] .	category 1: *company* category 2: *school* category 3: *artist* category 4: *sports* category 5: *politics, office* category 6: *transportation, car, bus, train* category 7: *building, construct, room, tower* category 8: *river, lake, mountain* category 9: *village* category 10: *animal, pet* category 11: *plant* category 12: *album* category 13: *film* category 14: *book, publication*	7
IMDB	x All in all, it was [MASK] .	positive: *good* negative: *bad*	500
Amazon	x All in all, it was [MASK] .	positive: *good* negative: *bad*	170

実験設定

- 利用したプロンプトやラベル名の例②

SST-2	x_1 It was [MASK] .	positive: *great* negative: *terrible*	9
MNLI	x_1 ? [MASK] , x_2	entailment: *yes* neutral: *maybe* contradiction: *no*	4
MNLI-mm	x_1 ? [MASK] , x_2	entailment: *yes* neutral: *maybe* contradiction: *no*	4
QNLI	x_1 ? [MASK] , x_2	entailment: *Yes, Indeed, Overall* not_entailment: *No, Well, However*	3
RTE	x_1 ? [MASK] , x_2	entailment: *Yes* not_entailment: *No*	10
MRPC	x_1 [MASK] , x_2	equivalent: *Yes* not_equivalent: *No*	9
QQP	x_1 [MASK] , x_2	equivalent: *Yes* not_equivalent: *No*	9
CoLA	x_1 This is [MASK] .	grammatical: *true* not_grammatical: *wrong*	7

実験設定：比較手法①

- ManualVerb

  ドメイン知識を持った人間の専門家がverbalizerを人手で定義

- Semantic Retrieval

  入力文の埋め込みとラベル説明文の類似度を測る

- NSP-BERT

  テキスト分類を含意関係認識と置き換えNext Sentence Predictionを利用して予測

- GPT-3/ChatGPT with descriptions

  タスクの説明をプロンプトに加えモデルに入力

AG News :
`[Descriptions]` Definition: In this task, you are given a sentence. Your job is to classify the following sentence into one of the four different categories. The categories are: “politics”, “sports”, “business”, and “technology”. Input: `[x]`. Output:

実験設定：比較手法②

- SimPTC
- LOTClass
- KPT
- Null Prompt

  [MASK]を文末に付与し直接ラベル名を推定
- Multi-Null prompting

  [MASK]を文頭と文末に付与し直接ラベルを推定

完全にZero-shotの手法

実験結果①：テキスト分類

- ゼロショットの手法において一番良い性能を記録

Human：人手の作業がある
KB：外部知識を利用
Unlabeld：ラベルのないコーパスを利用

Method	Human/KB	Unlabeled	AG News	DBPedia	IMDB	Amazon	Avg.
ManualVerb	✔	✗	$79.6_{0.6}$	$71.7_{1.1}$	$92.0_{0.7}$	$87.3_{0.4}$	82.7
Semantic Retrieval	✔	✗	$73.1_{1.2}$	$78.6_{0.8}$	$64.8_{1.3}$	$59.4_{0.7}$	69.0
NSP-BERT	✔	✗	$77.4_{0.6}$	$64.7_{5.3}$	$72.8_{1.1}$	$72.7_{3.9}$	71.9
GPT-3 w. descriptions	✔	✗	83.4	82.5	88.8	89.4	86.0
ChatGPT w. descriptions	✔	✗	83.8	92.0	92.7	**95.8**	91.1
SimPTC	✔	✗	$\mathbf{86.9}_{0.3}$	$\mathbf{93.2}_{1.0}$	$91.0_{0.0}$	$93.9_{0.0}$	**91.3**
LOTClass w/o. self train	✗	✔	82.2	86.0	80.2	85.3	83.4
LOTClass	✗	✔	86.4	91.1	86.5	91.6	88.9
KPT	✔	✔	86.7	87.4	**94.0**	94.6	90.7
Null Prompt	✗	✗	$67.9_{2.0}$	$56.8_{3.9}$	$82.5_{1.5}$	$89.4_{1.0}$	74.2
Multi-Null Prompt	✗	✗	$68.2_{1.8}$	$67.6_{1.8}$	$86.6_{0.6}$	$86.2_{2.7}$	77.2
NPPrompt	✗	✗	$\mathbf{85.2}_{0.5}$	$\mathbf{86.8}_{0.1}$	$\mathbf{94.2}_{0.2}$	$\mathbf{93.9}_{0.0}$	**90.0**

実験結果①：テキスト分類

- 多くのベースラインより上回る結果が得られたが、少しだけ最高性能を下回った

Method	Human/KB	Unlabeled	AG News	DBPedia	IMDB	Amazon	Avg.
ManualVerb	✔	✗	$79.6_{0.6}$	$71.7_{1.1}$	$92.0_{0.7}$	$87.3_{0.4}$	82.7
Semantic Retrieval	✔	✗	$73.1_{1.2}$	$78.6_{0.8}$	$64.8_{1.3}$	$59.4_{0.7}$	69.0
NSP-BERT	✔	✗	$77.4_{0.6}$	$64.7_{5.3}$	$72.8_{1.1}$	$72.7_{3.9}$	71.9
GPT-3 w. descriptions	✔	✗	83.4	82.5	88.8	89.4	86.0
ChatGPT w. descriptions	✔	✗	83.8	92.0	92.7	**95.8**	91.1
SimPTC	✔	✗	$\mathbf{86.9}_{0.3}$	$\mathbf{93.2}_{1.0}$	$91.0_{0.0}$	$93.9_{0.0}$	**91.3**
LOTClass w/o. self train	✗	✔	82.2	86.0	80.2	85.3	83.4
LOTClass	✗	✔	86.4	91.1	86.5	91.6	88.9
KPT	✔	✔	86.7	87.4	**94.0**	94.6	90.7
Null Prompt	✗	✗	$67.9_{2.0}$	$56.8_{3.9}$	$82.5_{1.5}$	$89.4_{1.0}$	74.2
Multi-Null Prompt	✗	✗	$68.2_{1.8}$	$67.6_{1.8}$	$86.6_{0.6}$	$86.2_{2.7}$	77.2
NPPrompt	✗	✗	$\mathbf{85.2}_{0.5}$	$\mathbf{86.8}_{0.1}$	$\mathbf{94.2}_{0.2}$	$\mathbf{93.9}_{0.0}$	**90.0**

実験結果①：テキスト分類

- GPT-3(175B)のような大規模なモデルよりも小さいモデルで高い性能を示した

Method	Human/KB	Unlabeled	AG News	DBPedia	IMDB	Amazon	Avg.
ManualVerb	✔	✗	$79.6_{0.6}$	$71.7_{1.1}$	$92.0_{0.7}$	$87.3_{0.4}$	82.7
Semantic Retrieval	✔	✗	$73.1_{1.2}$	$78.6_{0.8}$	$64.8_{1.3}$	$59.4_{0.7}$	69.0
NSP-BERT	✔	✗	$77.4_{0.6}$	$64.7_{5.3}$	$72.8_{1.1}$	$72.7_{3.9}$	71.9
GPT-3 w. descriptions	✔	✗	83.4	82.5	88.8	89.4	86.0
ChatGPT w. descriptions	✔	✗	83.8	92.0	92.7	**95.8**	91.1
SimPTC	✔	✗	$\mathbf{86.9}_{0.3}$	$\mathbf{93.2}_{1.0}$	$91.0_{0.0}$	$93.9_{0.0}$	**91.3**
LOTClass w/o. self train	✗	✔	82.2	86.0	80.2	85.3	83.4
LOTClass	✗	✔	86.4	91.1	86.5	91.6	88.9
KPT	✔	✔	86.7	87.4	**94.0**	94.6	90.7
Null Prompt	✗	✗	$67.9_{2.0}$	$56.8_{3.9}$	$82.5_{1.5}$	$89.4_{1.0}$	74.2
Multi-Null Prompt	✗	✗	$68.2_{1.8}$	$67.6_{1.8}$	$86.6_{0.6}$	$86.2_{2.7}$	77.2
NPPrompt	✗	✗	$\mathbf{85.2}_{0.5}$	$\mathbf{86.8}_{0.1}$	$\mathbf{94.2}_{0.2}$	$\mathbf{93.9}_{0.0}$	**90.0**

実験結果②：GLUEベンチマーク

- Zero-shotの手法の性能を上回る性能を示した

	MNLI (acc)	MNLI-mm (acc)	SST-2 (acc)	QNLI (acc)	RTE (acc)	MRPC (F1)	QQP (F1)	CoLA (Matt.)	Avg.
With human designed prompts / few-shot data									
Manual Label	50.8	51.7	83.6	50.8	51.3	61.9	49.7	2.0	50.2
In-context learning	**52.0**$_{0.7}$	**53.4**$_{0.6}$	84.8$_{1.3}$	53.8$_{0.4}$	60.4$_{1.4}$	45.7$_{6.0}$	36.1$_{5.2}$	−1.5$_{2.4}$	48.1
Auto-L	41.6$_{5.4}$	42.3$_{6.2}$	84.3$_{3.3}$	57.9$_{3.9}$	**61.9**$_{7.5}$	**67.7**$_{7.9}$	55.5$_{5.0}$	1.2$_{4.8}$	51.6
AMuLaP	50.8$_{2.1}$	52.3$_{1.8}$	**86.9**$_{1.6}$	53.1$_{2.8}$	58.9$_{7.9}$	56.3$_{5.0}$	60.2$_{2.7}$	2.3$_{1.4}$	52.6
Few-shot fine-tuning	45.8$_{6.4}$	47.8$_{6.8}$	81.4$_{3.8}$	**60.2**$_{6.5}$	54.4$_{3.9}$	76.6$_{2.5}$	**60.7**$_{4.3}$	**33.9**$_{14.3}$	**57.6**
Fully zero-shot									
Majority	32.7	33.0	50.9	49.5	52.7	**81.2**	0.0	0.0	37.5
Null Prompt	33.1$_{0.4}$	33.8$_{0.5}$	79.1$_{4.0}$	50.7$_{0.1}$	47.2$_{0.6}$	12.9$_{7.0}$	1.3$_{1.0}$	−1.1$_{2.0}$	32.1
Multi-Null Prompt	38.0$_{3.5}$	38.5$_{4.1}$	70.2$_{7.7}$	52.2$_{1.7}$	53.0$_{2.2}$	19.9$_{8.7}$	25.5$_{13.4}$	**6.2**$_{2.0}$	37.9
NPPrompt	**45.7**$_{0.6}$	**45.9**$_{0.5}$	**86.3**$_{1.2}$	**57.6**$_{0.7}$	**55.0**$_{3.4}$	79.8$_{1.6}$	**52.4**$_{0.4}$	4.9$_{4.1}$	**53.5**

実験結果②：GLUEベンチマーク

- タスクの平均で見るとfew-shotの手法も上回る性能

	MNLI (acc)	MNLI-mm (acc)	SST-2 (acc)	QNLI (acc)	RTE (acc)	MRPC (F1)	QQP (F1)	CoLA (Matt.)	Avg.
With human designed prompts / few-shot data									
Manual Label	50.8	51.7	83.6	50.8	51.3	61.9	49.7	2.0	50.2
In-context learning	**52.0**$_{0.7}$	**53.4**$_{0.6}$	84.8$_{1.3}$	53.8$_{0.4}$	60.4$_{1.4}$	45.7$_{6.0}$	36.1$_{5.2}$	−1.5$_{2.4}$	48.1
Auto-L	41.6$_{5.4}$	42.3$_{6.2}$	84.3$_{3.3}$	57.9$_{3.9}$	**61.9**$_{7.5}$	**67.7**$_{7.9}$	55.5$_{5.0}$	1.2$_{4.8}$	51.6
AMuLaP	50.8$_{2.1}$	52.3$_{1.8}$	**86.9**$_{1.6}$	53.1$_{2.8}$	58.9$_{7.9}$	56.3$_{5.0}$	60.2$_{2.7}$	2.3$_{1.4}$	52.6
Few-shot fine-tuning	45.8$_{6.4}$	47.8$_{6.8}$	81.4$_{3.8}$	**60.2**$_{6.5}$	54.4$_{3.9}$	76.6$_{2.5}$	**60.7**$_{4.3}$	**33.9**$_{14.3}$	**57.6**
Fully zero-shot									
Majority	32.7	33.0	50.9	49.5	52.7	**81.2**	0.0	0.0	37.5
Null Prompt	33.1$_{0.4}$	33.8$_{0.5}$	79.1$_{4.0}$	50.7$_{0.1}$	47.2$_{0.6}$	12.9$_{7.0}$	1.3$_{1.0}$	−1.1$_{2.0}$	32.1
Multi-Null Prompt	38.0$_{3.5}$	38.5$_{4.1}$	70.2$_{7.7}$	52.2$_{1.7}$	53.0$_{2.2}$	19.9$_{8.7}$	25.5$_{13.4}$	**6.2**$_{2.0}$	37.9
NPPrompt	**45.7**$_{0.6}$	**45.9**$_{0.5}$	**86.3**$_{1.2}$	**57.6**$_{0.7}$	**55.0**$_{3.4}$	79.8$_{1.6}$	**52.4**$_{0.4}$	4.9$_{4.1}$	**53.5**

考察

1. 重み付け関数と類似度関数による性能差
2. ロジットと確率の性能差
3. k-近傍の単語数による性能差
4. 言語モデルによる性能差
5. テキスト分類以外へのタスクへの適用

考察①：重み付け関数と類似度関数による違い

- 類似度関数をコサイン類似度に固定し重み付け関数を変更

$$\mathrm{Q}(y_j|x) = \sum_{v_i \in M(y_j)} w(v_i, y_j) \cdot \Theta([MASK] = v_i | x_{prompt} = \tau(x))$$

1. $Default \quad = \mathrm{w}(v_i, y_j) = \frac{\exp(S(emp(v_i), emb(y_j)))}{\sum_{v_i \in M(y_j)} \exp(S(emp(v_i), emb(y_j)))}$

2. $Same\ weight \quad = \mathrm{w}(v_i, y_j) = 1$

3. $average\ weight = \mathrm{w}(v_i, y_j) = \frac{S(emp(v_i), emb(y_j))}{\sum_{v_i \in M(y_j)} S(emp(v_i), emb(y_j))}$

考察①：重み付け関数と類似度関数による違い

- 重み付け関数による大きな性能差のない結果

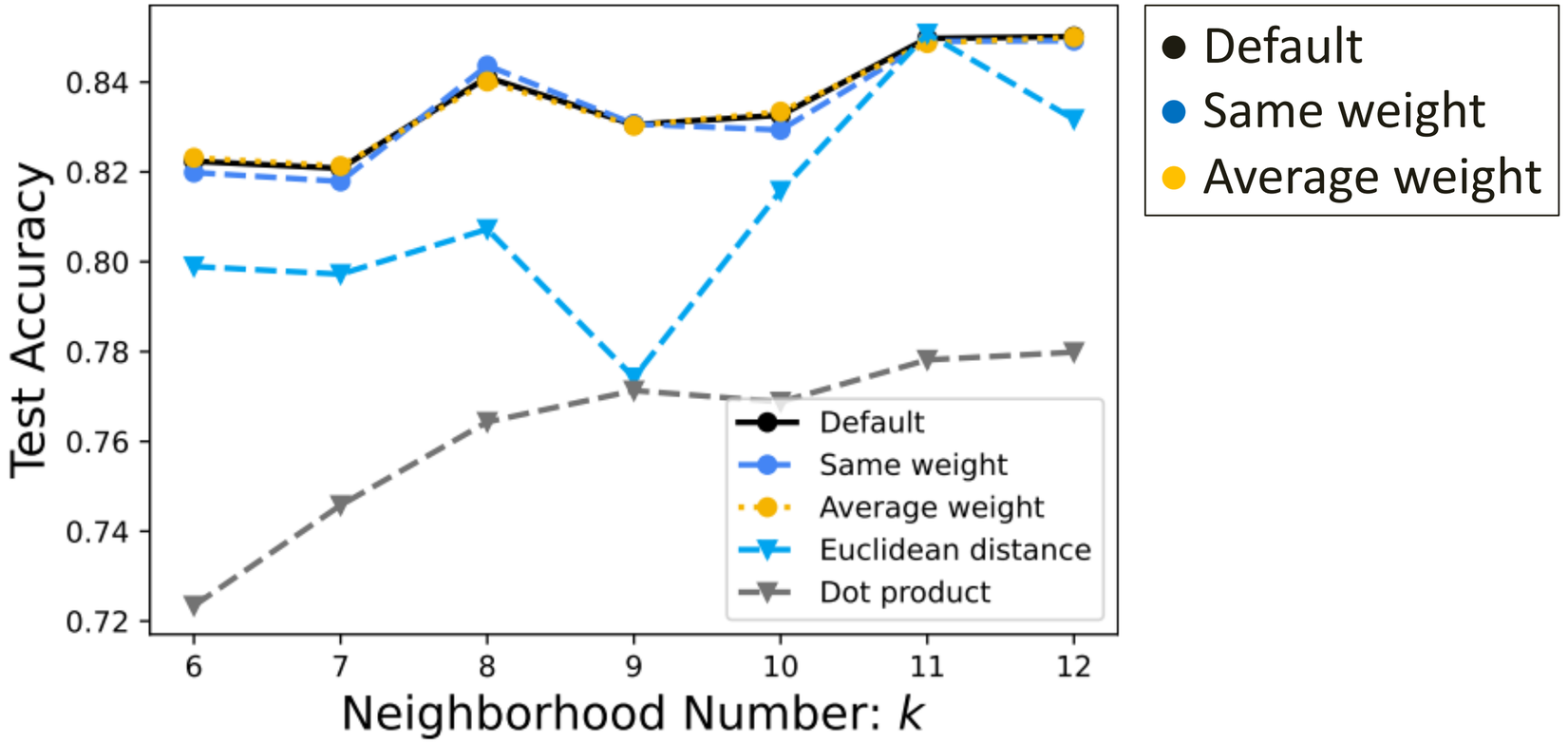


Figure 2: Effects of different aggregation.

考察①：重み付け関数と類似度関数による違い

- 重み付け関数を$\mathrm{w}(v_i, y_j) = 1$に固定し類似度関数を変更

1. $cosine\ similarity \quad = S\big(emb(v_i), emb(y_i)\big) = \frac{emb(v_i)}{||emb(v_i)||} \cdot \frac{emb(y_i)}{||emb(y_i)||}$

2. $Euclidean\ distance = S\big(emb(v_i), emb(y_i)\big) = -\big\|emb(v_i) - emb\big(y_j\big)\big\|$

3. $Dot\ product \quad = S\big(emb(v_i), emb(y_i)\big) = emb(v_i) \cdot emb\big(y_j\big)$

考察①：重み付け関数と類似度関数による違い

- 類似度関数によっては大きく性能に差が出る結果

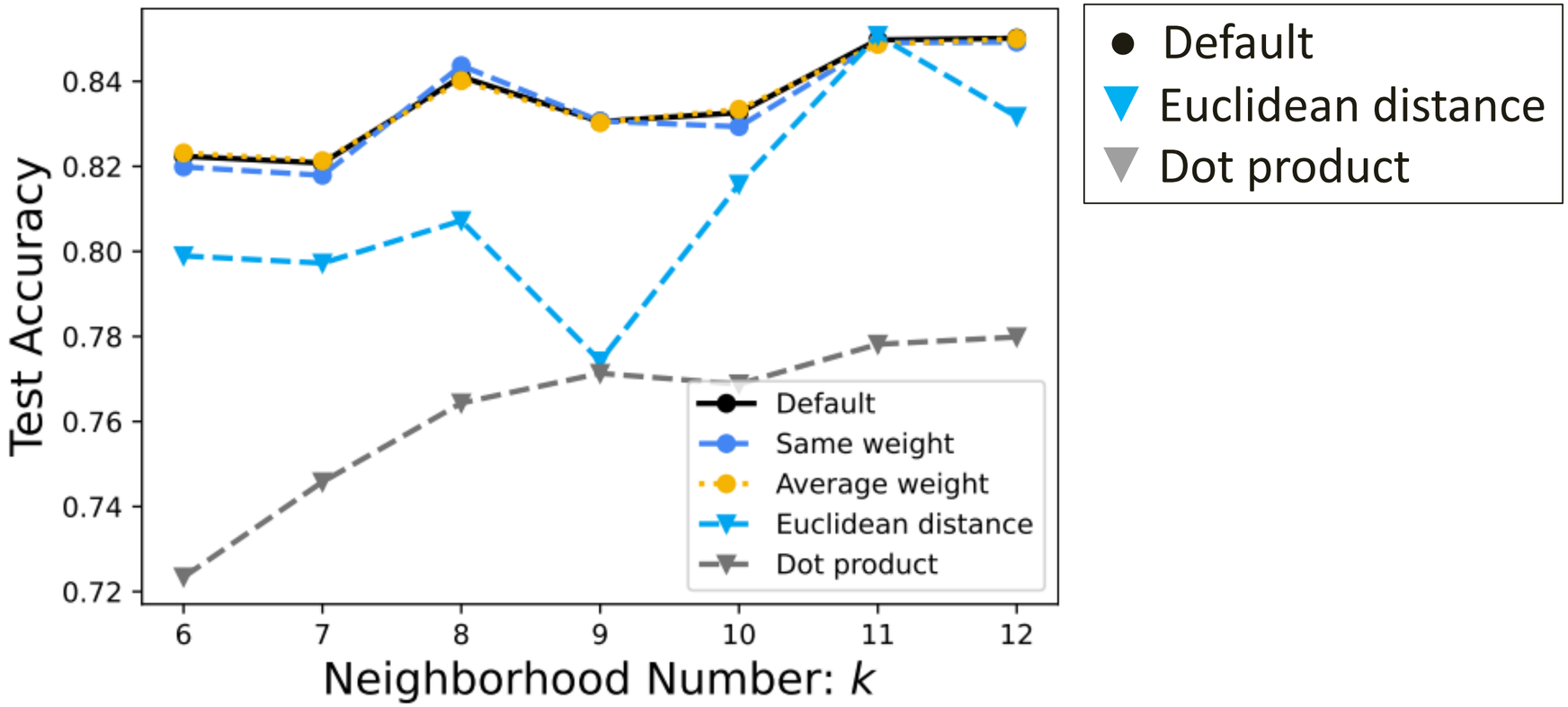
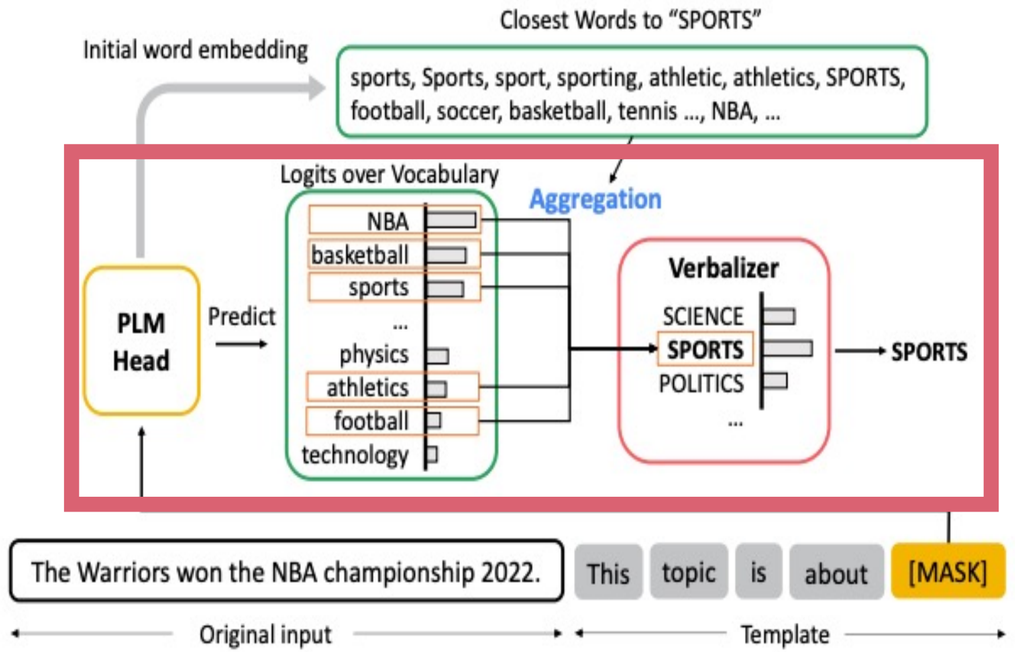


Figure 2: Effects of different aggregation.

考察②：ロジットと確率の性能差

- 近傍の単語数が少ない場合”sum logit”、多い場合は”sum prob”の性能が良い
- 一番高いスコアはsum logitの k=12 の時



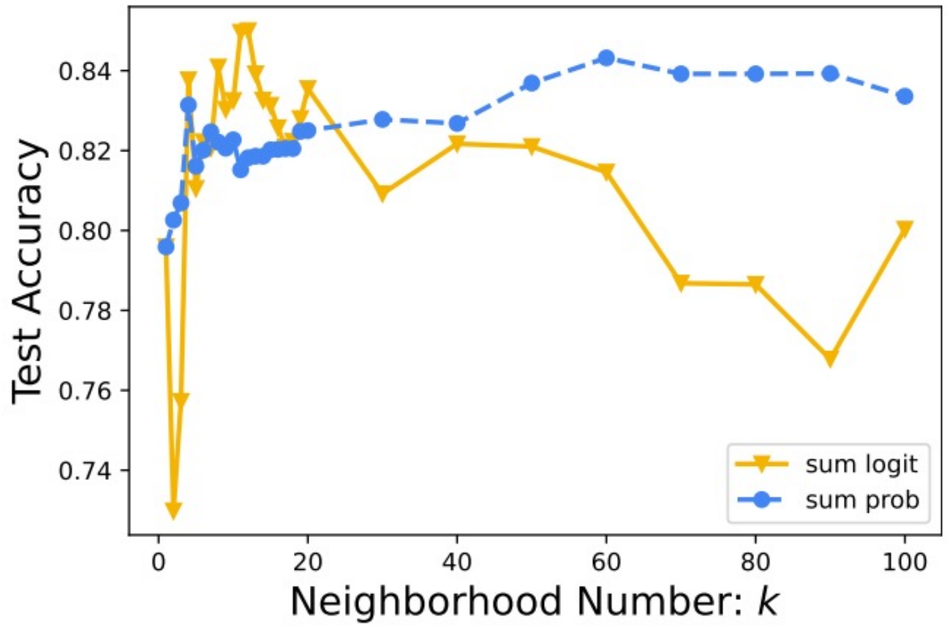
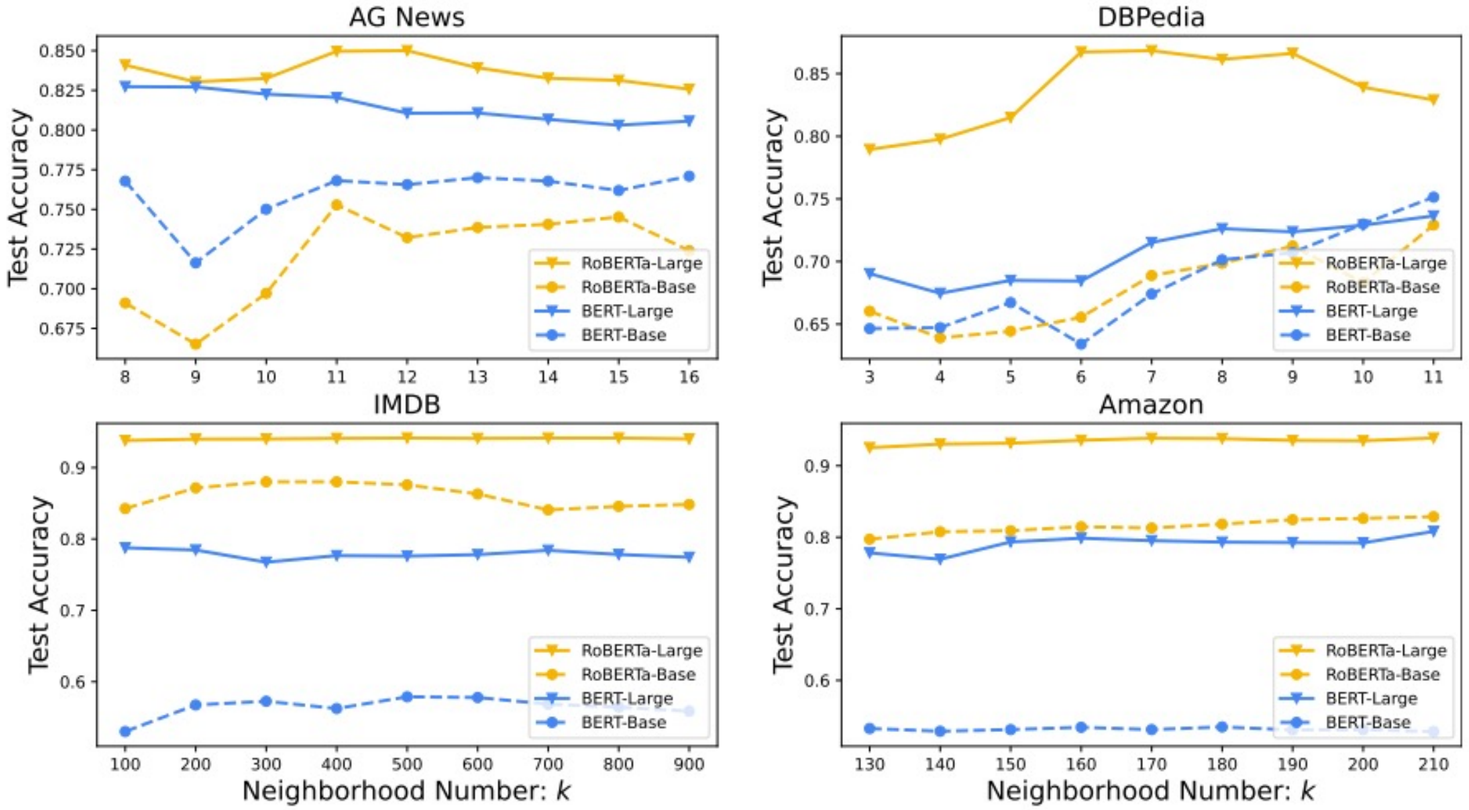


Figure 3: Test results on AG News.

考察③：k-近傍の単語数による性能差

- データセットによってkの値で性能差が生じる
- kを８～１０くらいの設定するのが良い



考察④：言語モデルの違いによる性能差

- 言語モデルの違いによる性能差を評価
  - T5-base, GPT2-base, BERT-base/large, RoBERTa-base/largeを比較
  - T5、GPTはモデルが最初に生成した単語を元にラベルを予測

考察④：言語モデルの違いによる性能差

- 言語モデルの違いで大きく性能に差が存在
  - PLMの初期単語埋め込みと語彙に依存
  - largeモデルはbaseモデルを上回る性能を示す

Method	AG	DB	IM	AZ	Avg.
NPPrompt-T5-base	76.8	78.3	68.5	65.3	72.2
NPPrompt-GPT2-base	81.1	78.1	83.7	85.6	82.1
NPPrompt-BERT-base	79.4	77.8	57.7	53.5	67.1
NPPrompt-BERT-large	82.7	80.9	81.6	80.8	81.5
NPPrompt-RoBERTa-base	75.3	82.8	88.7	83.9	82.7
NPPrompt-RoBERTa-large	**85.0**	**86.8**	**94.1**	**93.9**	**90.0**

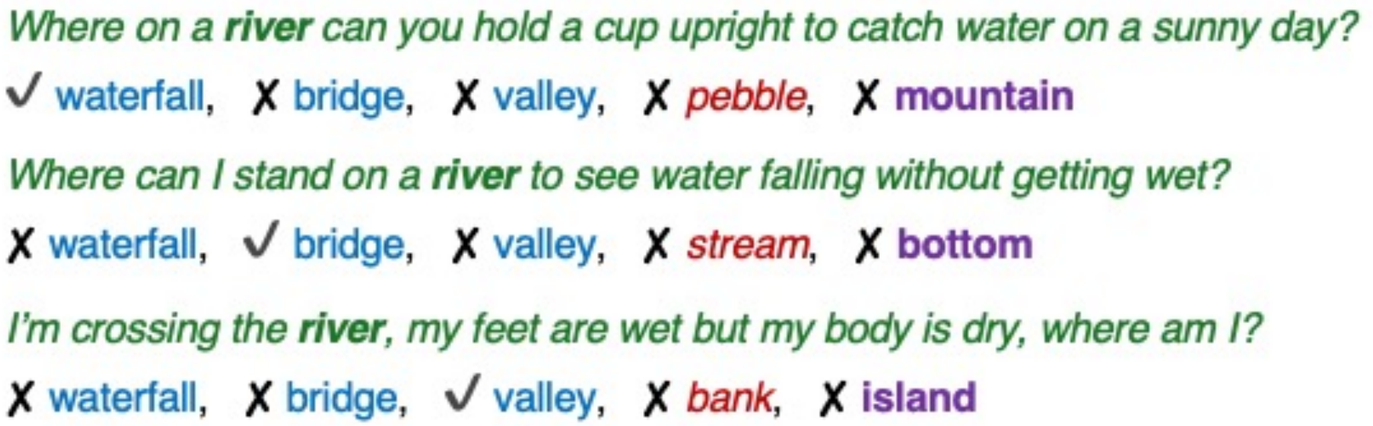
考察④：言語モデルの違いによる性能差

- 言語モデルの違いで大きく性能に差が存在
  - PLMの初期単語埋め込みと語彙に依存
  - largeモデルはbaseモデルを上回る性能を示す

Method	AG	DB	IM	AZ	Avg.
NPPrompt-T5-base	76.8	78.3	68.5	65.3	72.2
NPPrompt-GPT2-base	81.1	78.1	83.7	85.6	82.1
NPPrompt-BERT-base	79.4	77.8	57.7	53.5	67.1
NPPrompt-BERT-large	82.7	80.9	81.6	80.8	81.5
NPPrompt-RoBERTa-base	75.3	82.8	88.7	83.9	82.7
NPPrompt-RoBERTa-large	**85.0**	**86.8**	**94.1**	**93.9**	**90.0**

考察⑤：テキスト分類以外のタスクへの適用

- 質問応答へNPPromptを適用
  - データセットは選択問題 (CommonsenseQA)
  - “x The answer is [MASK].” というプロンプト
  - 比較モデル
    1. GPT-J Direct / CoT
    2. LaMDA (137B)



[Talmor+,19]

考察⑤：テキスト分類以外のタスクへの適用

- 質問応答へNPPromptを適用
  - QAでもある程度の性能
  - Few-shotのGPT-Jのスコアにも勝るような性能
  - 他のNLPタスクへも応用できる可能性

Method	CQA Dev Set Accuracy
Few-shot Direct GPT-J	20.9
Few-shot CoT GPT-J	36.6
Few-shot CoT LaMDA 137B	55.6
NPPrompt-RoBERTa-large	34.2

まとめと今後の展望

- Zero-shotでテキスト分類などの手法の性能を向上させる手法の提案
- PLMの語彙を利用することで人手のコストや追加データの用意が不要
- 従来のZero-shotの性能を上回る結果
- Few-shotの手法にも匹敵するような性能

- 今後の展望
  - ・キーワードの設計次第でラベル名が意味を持たなくても利用可能
  - ・この手法をより大きなモデルに適用したら性能は上がりそう