Vol.2020-NL-246 No.13
Vol.2020-SLP-134 No.13
2020/12/2

AMR複文構文パターン辞書作成および意味的曖昧性解消実験

山元 勇輝[1,a)]　松本 裕治[2,b)]　渡辺 太郎[1,c)]

概要：英語の複文を対象に Abstract Meaning Representation (AMR) と対応した構文パターン辞書を人手で作成し, 対象となる文を構文パターンと照合することで, 対応する AMR のフレームを出力するシステムを実装した. また, 一部の構文が複数の AMR 構造と対応する意味的曖昧性を解消するため, AMR コーパスと Wikipedia コーパスを利用して分類器を学習し, 評価実験を実施した. 本稿では, 追加検証とともにその結果と分析について報告する.

Complex Sentence Pattern Lexicon for AMR and Experiments on Semantic Ambiguity Resolution

1. はじめに

Abstract Meaning Representation (AMR) Parsing [1] における課題の一つとして, 複文構文の解析があげられる. 複文とは主節と一つ以上の従属節からなる文の分類であり, 単文や重文に比べ, 自然言語には数多くの種類が存在する. AMR では一つの動詞の意味を表すために PropBank [2] の述語項フレームを利用している [3]. 一方, 主節と従属節の動詞間の意味関係については整理されていない. 例外的に, 最新の AMR コーパスでは Bonial ら [4] に従い, 構文文法論的アプローチを取り入れているため **the X-er, the Y-er** のような比較構文の一部については図 1 のような構文フレームを導入している. しかし, 複文構文の観点からするとカバレッジが十分であるとはいえない.

複文構文の解析に取り組む上で我々が効果的であると考えるアプローチとして, まず文内に存在するパターンを特定し, 主節と従属節の動詞間の関係を表す情報 (以降, “relational AMR”) を Parser に与えるということが考えられる. 本研究では, その第一段階として, 英語の複文構文のパターンとそれに対応する relational AMR を含んだ辞書を作成し, パターン照合器も用意する. 特に AMR の枠組みにおいて, 我々の知る限り複文を対象とした網羅的な言語資源を作成する試みはなされていない.

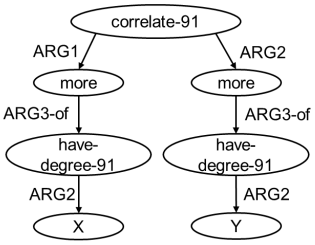

図 1　the X-er, the Y-er に対応する構文フレーム

パターンの中には, 複文構文に対応する relational AMR が意味的曖昧性を持つものも存在する. このような例についてはパターン照合後に意味的曖昧性の解消を行う必要があるため, 本研究では次の段階として多クラス分類器の学習を行う. しかし, 最新の AMR コーパスでもデータ数は 59.2K であり, 該当の意味ラベルを持つ用例のみ利用することを考慮すると, 本タスクの学習データとして利用するには規模が小さい. そこで, 大規模データから学習データを獲得し, 分類器の性能を向上させることを目指す. また, いくつかの追加検証を実施し, 大規模データをより効果的に利用する方法を模索する.

[1] 奈良先端科学技術大学院大学
[2] 理化学研究所革新知能統合研究センター
[a)] yamamoto.yuki.yt0@is.naist.jp
[b)] yuji.matsumoto@riken.jp
[c)] taro@is.naist.jp

ⓒ 2020 Information Processing Society of Japan

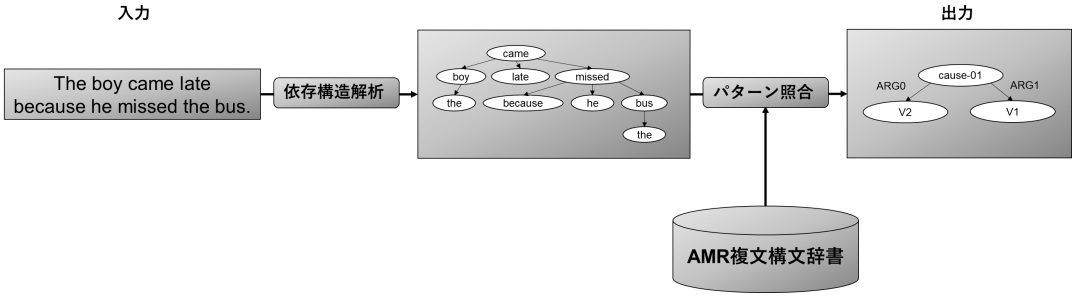

図 2 AMR 複文構文辞書を用いたパターン照合

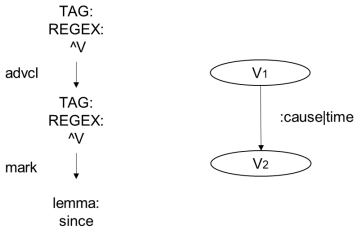

図 3 V_1 since V_2 のパターンおよび対応する relational AMR

2. AMR 複文構文辞書の作成

AMR 複文構文辞書に登録するパターンは Universal Dependencies (UD) 形式の依存構造を基礎構造として利用する. ただし, 依存構造だけでは目的外の文も照合する可能性があるため, 必要に応じて木構造のノードに POS タグ, lemma あるいは正規表現を, エッジには依存構造ラベルを利用し, 語彙的情報と構文的情報を併せ持つパターンとして記述する. なお, 本研究ではパターン照合器として, spaCy の Dependency Matching ツールを利用する. このツールは UD 形式の木構造に対して照合を行うため, 対象の文については Stanza の依存構造解析器を利用する.

3. 構文の意味的曖昧性

複文構文の中には同じ構文であるにもかかわらず意味的に異なる複数の用法を持つものが存在する [5]. 例えば, 図 3 で表した **V_1 since V_2** という構文は以下の 2 通りの用法がある. (V_1, V_2 はそれぞれ主節, 従属節の主動詞を指す.)

(1) a. She was in bed *since* she had a cold. (因果)
b. He had lived here *since* he was five. (時間)

V_1 when V_2 は, 条件と時間の用法を持つ.

(2) a. She was in bed *when* she had a cold. (条件)
b. He had lived here *when* he was five. (時間)

分詞構文 **V_2-ing, V_1** は以下の 4 用法の曖昧性を持つ.

(3) a. Not *speaking* the local language, he tried to communicate with sign language. (因果)
b. *Seeing* that first flying machine, no one could have dreamed of today's airplanes. (条件)
c. *Intending* to diet, she still eats cake after every meal. (譲歩)
d. *Having* tried to ring him ten times that day, I gave up and wrote him a mail. (時間)

このように, 辞書内に存在する複文構文で複数の用法を持つものは, 因果, 条件, 譲歩, 時間のうちいずれかの用法の組み合わせをもつことが辞書作成の過程でわかっている. 意味的曖昧性を辞書に記述する方法としては, 一つの構文パターンに対して一つの relational AMR 内に OR 演算子 | を用いて候補を記述しておくことで, 辞書の登録数が不必要に大きくなることを防ぐことができる. 一方で, パターン照合後, 複数の用法の候補から一つを決定する必要があるため, 本研究では上述の 4 用法について分類器を学習する.

4. 実験

本実験では, 意味的曖昧性をもつ構文にも正解の意味ラベルが記述されている AMR コーパス内の文のみで学習した場合と比較して, ラベル付けされていない大規模データの利用が効果的であるのかを検証する.

4.1 文ペア分類タスク

本実験では事前学習モデルである BERT に文ペア分類タスクで Fine-Tuning を行う. その際, 因果, 条件, 譲歩, 時間の用法にそれぞれ CAUSE, COND, CONC, TIME というクラスを割り当て, 4 クラス設定の文ペア分類を目標とする. 文ペア分類では連続した 2 文を入力として, その文間の関係を表すクラスを出力する. このような入力が可能であるのは, BERT が事前学習の段階において, 入力された 2 文が連続しているか否かを解く Next Sentence Prediction のタスクで学習されているからである. 我々は連続した文ペアではなく, 従属節からなる文と主節からなる文のペアを利用する.

4.2 データセットの作成

学習とテストに利用するデータを作成するため, AMR コーパスと Wikipedia コーパスを利用する. 前者の最新版である AMR Release 3.0 は合計 59,255 の文と AMR グラフのペアを含んだコーパスであり, すべて人手でアノテートされたものである. 後者は約 20 億単語を含んだ大規模

表 1 AMR コーパスから獲得した Gold データの内訳

	従属接続詞	Total
CAUSE	*because, since*	544
COND	*if, unless*	1,154
CONC	*though, although, even if*	107
TIME	*once, when, while, as*	1,122

表 2 Wikipedia コーパスから獲得した Silver データの内訳

	従属接続詞	Total
CAUSE	*because*	27,264
COND	*if, unless*	36,595
CONC	*though, although, even if*	53,721
TIME	*once, whenever*	23,483

コーパスである.

まず, これらのコーパスから上述の 4 用法のうちいずれかをもつ複文を発見する必要がある. そのため, Stanza の依存構造解析器をコーパス内の文に利用し, いくつかの従属接続詞を選択して $\mathbf{V}_1$ **従属接続詞** $\mathbf{V}_2$ の構造でパターン照合する. これにより, 例えば $\mathbf{V}_1$ **because** $\mathbf{V}_2$ と照合した文には CAUSE のクラスを割り当てることができる. ここで, AMR コーパスについては文と AMR グラフのアラインメントを参考にできるので, 曖昧性のある従属接続詞をもつ用例も獲得できる. 例えば, $\mathbf{V}_1$ **since** $\mathbf{V}_2$ の構造を持ちつつ, AMR グラフの対応部分に:cause の意味ラベルを持つものは CAUSE クラスの用例として利用する. 一方, Wikipedia コーパスについては正解が不明のため, 意味的曖昧性をもつ従属接続詞を含む文は対象外とする.

次に, 発見した依存構造木を主節と従属節で分割された文ペアの形式に変換する. 依存構造木のなかで, V_1 および V_2 が子に持つ要素を探索する. ここで, V_2 以下の要素は従属節に当たる部分であり, V_1 以下の要素から従属節の要素を除外すれば主節が得られる. 最後に ID 順に線形化することで主節あるいは従属節からなる文を得ることができる. なお, 従属節に関しては文の形式にするために先頭の従属接続詞を除外する.

以降, AMR コーパスと Wikipedia コーパスがもとのデータをそれぞれ Gold データ (gold standard), Silver データ (silver standard) と呼ぶ. また, 簡単のため Gold データで学習したモデルを Gold とし, Silver データで学習したモデルを Silver と表記する.

4.3 実験設定

本実験では Gold をベースラインとし, Silver との評価を比較する. 表 1 のとおり Gold データの規模が小さいため, ホールドアウト法によるテストデータの分割では偏りが生じることが懸念される. そこで, Gold データによる学習については汎用的な性能の確認のために 5-分割交差検証を行う. 公平のため, Silver の評価に際しても交差検証のために 5 分割した Gold データをテストデータとして利用し, そのスコアの平均値を報告する. また, Silver データ中のノイズがスコアに影響を及ぼす可能性を考慮して, Silver の学習に用いるシード値を 5 回変更し, その平均値を最終スコアとして利用する. なお学習における主要なハイパーパラメータは表 3 の記載に従うものとする. ただし, バッチサイズに関しては各データサイズを考慮して Gold データは 16, Silver データ利用時は 64 とする.

表 3 主要なハイパーパラメータの設定

ハイパーパラメータ	値
epoch	10
optimizer	Adam
adam epsilon	1e-8
learning rate	4e-5

4.4 評価指標

本節では [6] に基いて, 多クラス分類タスクで利用される評価指標について説明する.

各クラス $c \in C$ をラベルとして持つテストデータの集合 X が与えられたとき, 各サンプル $x \in X$ について正解クラスを c_x, 予測クラスを $\hat{c}_x$ と表すと, クラスごとの *precision*, *recall*, *F*-score は以下のように計算される.

$$P(c) = \frac{|\{x \in X | c_x = c \wedge c_x = \hat{c}_x\}|}{|\{x \in X | \hat{c}_x = c\}|} \quad (1)$$

$$R(c) = \frac{|\{x \in X | c_x = c \wedge c_x = \hat{c}_x\}|}{|\{x \in X | c_x = c\}|} \quad (2)$$

$$F(c) = \frac{2 \cdot P(c) \cdot R(c)}{P(c) + R(c)} \quad (3)$$

マクロ平均は, 各クラスでの予測結果に均等に重み付けをして計算される.

$$F_{macro} = \frac{1}{|C|} \cdot \sum_{c \in C} F(c) \quad (4)$$

マイクロ平均は X のすべてのサンプルについての予測結果を割り出してから計算される.

$$P_{micro} = R_{micro} = \frac{|\{x \in X | c_x = \hat{c}_x\}|}{|\{x \in X\}|} \quad (5)$$

$$F_{micro} = \frac{2 \cdot P_{micro} \cdot R_{micro}}{P_{micro} + R_{micro}} = P_{micro} = R_{micro} \quad (6)$$

本実験では, モデル全体の性能を求めるためにこれら両方の指標を採用して評価を行う.

5. 結果

表 4 から F のマクロ平均およびミクロ平均ともに Gold で学習した方が高いスコアが出ている. この結果から, 大規模データを単独で利用することは分類器の性能の向上に特に有効ではないといえる.

次に表 5, 6 でクラス別の評価をみると, CAUSE と COND

ⓒ 2020 Information Processing Society of Japan

表 4 全体のマクロ平均およびマイクロ平均評価

	P_{macro}	R_{macro}	F_{macro}	F_{micro}
Gold	.52 ± .01	.51 ± .02	.51 ± .02	.63 ± .06
Silver	.53 ± .02	.58 ± .01	.49 ± .01	.57 ± .00

表 5 Gold データ利用時のクラス別の評価

	P	R	F
CAUSE	.48 ± .02	.53 ± .06	.50 ± .03
COND	.72 ± .03	.72 ± .04	.72 ± .02
CONC	.16 ± .03	.08 ± .02	.10 ± .03
TIME	.72 ± .01	.73 ± .01	.72 ± .01

表 6 Silver データ利用時のクラス別の評価

	P	R	F
CAUSE	.46 ± .01	.54 ± .01	.50 ± .01
COND	.65 ± .00	.76 ± .01	.74 ± .01
CONC	.16 ± .01	.63 ± .03	.25 ± .01
TIME	.84 ± .01	.37 ± .00	.51 ± .01

については概ね同様のスコアであることがみて取れる．一方, Gold の R をみると CONC が極端に低く TIME が高いが, Silver ではその傾向が逆転している．ただし, CONC の F をみるといずれのモデルも低水準であり, F_{macro} のスコアを下げる要因となっている．

6. 追加検証と考察

本節では, 前節の結果を受けてさらに検証を行い, 分類器の性能の向上を模索する．以下の結果はすべて実験と同じ方法で評価を行う．

大規模データの追加による影響

実験では Gold と Silver のデータそれぞれについて単独で学習に利用したが, 本検証では両データを結合して利用した場合の評価を調査したい．そこで, より大きなデータサイズで学習した場合が有効であるのかを確認するため, Silver データを追加する量について, Gold に対する比率 r を変化させてその推移をみた．

表 7 によるとデータの追加によって次第に F が上昇している．小さいデータサイズを補うことで性能が向上していると考えられる．その一方で, さらにデータを追加していくと下降がみられる．これは, Gold データの要素が薄まり, 表 4 における Silver の評価に近づいているのだと考えられる．このことから, ある一定の割合で Silver を追加することが有効であることがわかった．

大規模データによる事前の Fine-Tuning の影響

4.1 節で述べたように BERT に文ペアを入力として Fine-Tuning に利用することができるのは, Next Sentence Prediction のタスクによって, 2 文間に前後関係があるか否かを事前学習したモデルであるからである．一方, 本研究における Fine-Tuning の目的を考えると, 主節からなる文と従属節からなる文の論理関係について事前学習モデルが学習していることが望ましい．そこで, Gold データによる Fine-Tuning に先行して Silver データで BERT を Fine-Tuning した場合の検証を行った．

表 7 Silver データの追加によるマクロ平均およびマイクロ平均の推移

r	P_{macro}	R_{macro}	F_{macro}	F_{micro}
0	.52 ± .01	.51 ± .02	.51 ± .02	.63 ± .06
1	.54 ± .02	.55 ± .03	.54 ± .02	.66 ± .01
2	.54 ± .01	.56 ± .02	.54 ± .01	.64 ± .01
3	.55 ± .02	.59 ± .02	.56 ± .02	.65 ± .01
4	.53 ± .03	.57 ± .04	.54 ± .03	.63 ± .03
5	.54 ± .02	.58 ± .03	.54 ± .03	.63 ± .02

表 8 事前に Silver データで Fine-Tuning した場合のマクロ平均およびマイクロ平均評価

	P_{macro}	R_{macro}	F_{macro}	F_{micro}
FT	.61 ± .02	.60 ± .03	.60 ± .03	.69 ± .01

表 9 事前に Silver データで Fine-Tuning した場合の評価

	P	R	F
CAUSE	.58 ± .02	.56 ± .06	.57 ± .03
COND	.72 ± .02	.77 ± .02	.75 ± .02
CONC	.39 ± .08	.36 ± .15	.37 ± .11
TIME	.74 ± .02	.72 ± .03	.73 ± .02

表 10 学習データにおいて従属接続詞を明示した場合のマクロ平均およびマイクロ平均評価

	P_{macro}	R_{macro}	F_{macro}	F_{micro}
Gold+s	.63 ± .05	.61 ± .01	.60 ± .01	.75 ± .02
FT+s	.67 ± .04	.67 ± .03	.67 ± .04	.76 ± .02

FT は事前に Silver データで Fine-Tuning したモデルであることを表す．結果は表 8, 9 のとおり全クラスにおいて Gold の F を上回っており, 特に F_{macro} で.09 の上昇がみられたことから有効な手法であるといえる．

従属接続詞の明示化の影響

4.2 節で作成した学習データは従属接続詞を削除して得た文のペアであるが, 従属接続詞を従属節の文から削除せずに学習に利用することでその性能への影響をみた．

ここで, +s は学習データに従属接続詞を含むことを表す．表 10 の Gold+s をみると表 4 の Gold と比較して全体的な上昇がみられる．また, FT+s をみても表 8 の FT より F_{macro} がさらに.07 上昇しており, CAUSE に関しては表 9 で.57 であった F が.76 まで上がった．以上から, 学習データ内での従属接続詞は非常に重要な役割を果たしているといえる．

クラス数の削減による影響

最後に, どのモデルにおいても CONC の評価が他のクラスの水準まで上がらないことを考慮し, CONC と COND を結合した CC+CD を用いて 3 クラス分類後, CONC と COND の 2 クラス分類を行う手法を試みた．

ⓒ 2020 Information Processing Society of Japan

表 11 学習データにおいて従属接続詞を明示した場合のクラス別の評価

		P	R	F
Gold+s	CAUSE	.63 ± .04	.88 ± .02	.74 ± .03
	COND	.81 ± .05	.70 ± .03	.75 ± .03
	CONC	.31 ± .19	.08 ± .04	.12 ± .05
	TIME	.77 ± .01	.79 ± .03	.79 ± .02
FT+s	CAUSE	.68 ± .05	.87 ± .03	.76 ± .03
	COND	.82 ± .02	.71 ± .04	.76 ± .03
	CONC	.41 ± .11	.30 ± .10	.34 ± .10
	TIME	.79 ± .02	.81 ± .03	.80 ± .02

表 12 3 クラス分類における全体のマクロ平均およびマイクロ平均評価

	P_{macro}	R_{macro}	F_{macro}	F_{micro}
Gold	.69 ± .02	.69 ± .02	.69 ± .02	.71 ± .01
Silver	.61 ± .02	.53 ± .04	.52 ± .00	.57 ± .02

表 13 3 クラス分類におけるクラス別の評価

		P	R	F
Gold	CAUSE	.58 ± .04	.58 ± .05	.58 ± .04
	CD+CC	.73 ± .02	.77 ± .02	.75 ± .01
	TIME	.77 ± .02	.72 ± .01	.74 ± .01
Silver	CAUSE	.45 ± .01	.44 ± .01	.44 ± .01
	CD+CC	.55 ± .00	.86 ± .01	.67 ± .00
	TIME	.83 ± .01	.30 ± .01	.45 ± .01

表 14 2 クラス分類における全体のマクロ平均およびマイクロ平均評価

	P_{macro}	R_{macro}	F_{macro}	F_{micro}
Gold	.86 ± .07	.73 ± .05	.77 ± .06	.94 ± .01
Silver	.65 ± .01	.76 ± .02	.68 ± .01	.86 ± .01

表 15 2 クラス分類におけるクラス別の評価

		P	R	F
Gold	COND	.95 ± .01	.99 ± .01	.97 ± .01
	CONC	.77 ± .13	.47 ± .10	.58 ± .11
Silver	COND	.96 ± .00	.88 ± .01	.92 ± .01
	CONC	.33 ± .02	.64 ± .03	.43 ± .02

表 13 を表 5 と比較すると, Silver では全体の性能が下がっているが, Gold においては CAUSE, TIME にも良い影響を及ぼしていると判断できる. また, 表 14 の平均スコアをみても高水準であり, Gold データで Fine-Tuning する場合においてその有効性を示している.

7. 関連研究

我々は複文構文辞書作成に際して, 意味的曖昧性がある構文パターンに対して候補となる構造を列挙するのではなく, それぞれにただ一つの relational AMR を与えており, 分類器の結果によって意味を最終的に決定するというアプローチをとっている. これは, 辞書の登録数をむやみに増やすのではなく中核部分だけを記述し, 実際の使用時に何らか生成的な操作によって意味を決定するという生成語彙論 [7] の立場と類似している.

追加検証では BERT に 2 文間の論理的関係を捉えさせるため, Silver データで事前に Fine-Tuning を行った. これと同様, 最終的な目的に Fine-Tuning するため, 事前に大規模データを利用して学習を行った研究が存在する. Xu ら [8] は AMR Parsing のタスクに Fine-Tuning するために, 機械翻訳など複数のタスクで事前学習を行っており, その有効性を報告している.

8. おわりに

本研究では, AMR Parsing に利用できる資源として複文構文を対象とした辞書を作成し, 一部の構文の意味的曖昧性の解消のために意味ラベルを予測する多クラス分類器を学習し, 評価実験を行った. 実験結果から, 既存の AMR コーパスのみを使用した場合と比較して, 大規模データを単体で利用した場合には性能がわずかに下回ることがわかった. しかし, 追加的に実施した検証の結果, 大規模データを AMR コーパスのデータに一定量追加したり, AMR コーパスで Fine-Tuning する前段階における Fine-Tuning で用いるなど, 補足的な形で活用することで性能が大幅に向上することがわかった. また, 学習データにおける従属接続詞の明示や, 場合によってはクラス数の削減を行うことが有効であることも明らかになった. 現在, 我々は複文構文を考慮した AMR Parser の実装に関する研究を進めており, 本研究の成果を活用する予定である.

参考文献

[1] Flanigan J., Thomson S., Carbonell J., Dyer C., and Smith N. A.; A Discriminative Graph-Based Parser for the Abstract Meaning Representation, In Proc. of ACL, 2014.
[2] Palmer M., Gildea D., and Kingsbury P.: The Proposition Bank: An Annotated Corpus of Semantic Roles. CL, 31(1), pp. 71-106, March, 2005.
[3] Banarescu L., Bonial C., Cai S., Georgescu M., Griffitt K., Hermjakob U., Knight K., Koehn P., Palmer M., and Schneider N.: Abstract Meaning Representation for Sembanking, In Proc. of LAW-ID, 2013.
[4] Bonial C., Badarau B., Griffitt K., Hermjakob U., Knight K., O'Gorman T., Palmer M., Schneider N: Abstract Meaning Representation of Constructions: The More We Include, the Better the Representation, In Proc. of LREC, 2018.
[5] 山口俊治：英語構文全解説，研究社 (2013).
[6] Damasch, M., Dönicke, T., and Lux, F.: Multiclass Text Classification on Unbalanced, Sparse and Noisy Data, In Proc. of the First NLPL Workshop on Deep Learning for Natural Language Processing, 2019.
[7] Pustejovsky J.: The Generative Lexicon, MIT Press (1998).
[8] Xu D., Li J., Zhu M., Zhang M., and Zhou G.: Improving AMR Parsing with Sequence-to-Sequence Pre-training, CoRR, 2020.

ⓒ 2020 Information Processing Society of Japan