# CNN を用いた交通教則からの交通用語間関係抽出

八木 智也　　三輪 誠　　佐々木 裕
豊田工業大学
{sd15092, makoto-miwa, yutaka.sasaki}@toyota-ti.ac.jp

## 1 はじめに

現在の自動運転のためのプログラムは交通法規や交通マナーなどの運転に必要な知識がコード上に組み込まれているため更新が困難である．自動運転技術の実現に向けて，必要な知識をテキストから抽出し，交通オントロジーとして整理する手法が提案されている [1]．このような交通オントロジーを用いることで，交通オントロジーを参照して動作する自動運転システムを実現し，交通法規の更新の際や交通法規の異なる国での自動運転システムの利用において，システムを改変することなく，交通オントロジーの変更のみで対応できるようになると考えられる．このオントロジーの自動構築には，人手では困難な量の作業が必要であるため，テキストからの情報を自動で抽出し，オントロジーの構築を補助するシステムが必要とされている．

交通オントロジーの構築には交通用語抽出とそれらの用語間の関係抽出が必要である．本研究ではこの内の関係抽出を対象とする．この交通用語間の関係抽出においては，河辺 [1] は Support Vector Machine（SVM）を用いた．現在，多くの関係抽出タスクにおいて深層学習による精度向上が報告されている．例えば，Zeng ら [2] は，英文からの用語間関係抽出において畳み込みニューラルネットワーク（Convolutional Neural Network; CNN）を用い，高い分類精度を達成している．この CNN は，言語処理においてよく用いられる再帰型ニューラルネットワーク（Recurrent Neural Network; RNN）を用いる場合に比べて，並列化が容易であるという利点もある． SVM などの従来の特徴を用いる手法では特徴エンジニアリングが重要であるが，CNN を用いることで問題に見合った特徴を単語および文レベルで自動的に抽出することが可能である．そこで，本研究では CNN を利用して交通用語の関係抽出を行い，交通用語間の関係をより高い精度で抽出することを目的とする．

## 2 関連研究

### 2.1 交通オントロジーの自動構築

河辺 [1] は国家公安委員会が公開している交通教則文書 [3] に用語・関係をタグ付けし，交通用語とその間の関係をまとめたアノテーションデータを作成した．そして用語抽出に条件付き確率場 (Conditional Random Fields; CRF) を，関係抽出に SVM を用いて，交通オントロジー構築のためのシステムを作成した．

河辺の関係抽出は文レベルの関係を対象に，1 文中の任意の 2 つの用語に着目したとき正解データにおいて関係あり，なしとなるペアがどのような文脈に出現しているのかを素性ベクトルで表現し，SVM で学習した．そして未知の文において関係を持つペアの素性ベクトルをもとに分類を行うことで関係抽出を行った．用いた素性を表 1 に示す．素性の値は 0 か 1 の 2 値のものと，0 から 1 に正規化された実数のものがある．素性に利用される構文木は交通教則文を CaboCha[4] により構文解析したものを用いている．

### 2.2 CNN

ニューラルネットワークは脳の情報処理システムを模倣して作られたモデルである．入力に重みに応じた非線形変換を繰り返し，理想値と出力値の誤差を小さくするよう重みを調整し，学習する．CNN はニューラルネットワーク構造の一つで，入力ベクトルのうちの一定の領域を一つの特徴として表現し，それを畳み込んで広い範囲の特徴を抽出し，分類する．CNN はそれぞれの領域の計算を独立に並列で計算できるため，RNN に比べて計算時間を短くできる．

### 2.3 fastText

fastText[5] は単語ベクトルを学習する手法の 1 つであり，単語を文字 $n$-gram に分解し，それぞれの $n$-

gram の表現ベクトルの合計でその単語ベクトルを表現する．これによって低頻度語に対しても質の良いベクトル表現を得られる．このようにして学習した単語ベクトルを初期値として使用することで，言語処理モデルの性能が向上することが知られている．

## 3 提案手法

本研究では，CNN をベースに，Verga ら [6] のモデルを参考にして，用語ペアの表現から求められるスコアをもとに関係を分類する手法を提案する．作成したモデルを図 1 に示す．

本手法ではまず，各単語に対応付けられた単語ベクトルから CNN によって文脈情報を各単語に畳み込んだ単語表現を獲得する．次に，入力中のそれぞれの用語について，その用語に含まれる各単語表現の和を取り，これに用語タイプのベクトル表現を結合して用語の中間表現 $h_i$ とする．さらにこの中間表現 $\boldsymbol{h}_i$ に次のような処理を行い，用語ペア中の前方としての用語表現 ($Head$), 後方としての用語表現 ($Tail$) を計算する．

$$\boldsymbol{e}_i^{Head} = W_{Head2}(\tanh(W_{Head1}\boldsymbol{h}_i)) \quad (1)$$

$$\boldsymbol{e}_i^{Tail} = W_{Tail2}(\tanh(W_{Tail1}\boldsymbol{h}_i)) \quad (2)$$

最後に，このようにして計算した用語表現に次の計算を行い，$i$ 番目の用語と $j$ 番目の用語間の関係の有無 $l$ のスコア $A_{ijl}$ を求める．

$$A_{ijl} = \boldsymbol{e}_i^{Head^\top} L_l \boldsymbol{e}_j^{Tail} \quad (3)$$

ただし 1 つの用語からそれ自身に向かう関係はないとしているため，$A_{iil}$ は計算しない．複数ペアの関係の計算は独立であるため同時に行うことができる．そのため入力中における用語ペアの組み合わせすべてについて関係のスコアを一度に求める事が可能となる．

本研究ではある関係 X ごとに「X がある」,「逆向きに X がある」「X の関係はない」の 3 種類のスコアを計算し，最も高いスコアとなったものを，関係 X についてのモデルの予測とする．

予測では図 2 のようにある方向で関係なしと判断されても逆の方向で関係があると判断されればモデルとして関係ありと判断する．また,「交通量」から「少ない」に向かって「STATE」,「少ない」から「交通量」に向かって「STATE」というように同じものが予測され関係の向きが判断できない場合は，簡単のため，文中に前に出現する用語から後に出現する用語に対して判断された関係を優先することとする．

表 1: 河辺 [1] で用いられた素性

| 素性名 | 素性の内容 |
|---|---|
| Word Distance | 2 用語間の単語数 |
| Structure Level | 構文木上での 2 用語の所属階層 |
| Structure Distance | 構文木上での 2 用語間係り受け数 |
| PreDot Phrase | 用語の前の句が読点か |
| Parentheses | 用語が括弧で囲まれているか |
| Parentheses-Verb | 括弧内に動詞を含むか |
| Near PosTag | 用語の前後の品詞 |
| Case | 直後にある助詞の種類（格） |
| Probability | 用語の生起確率 |
| N-gram | 4 範囲での N グラム |

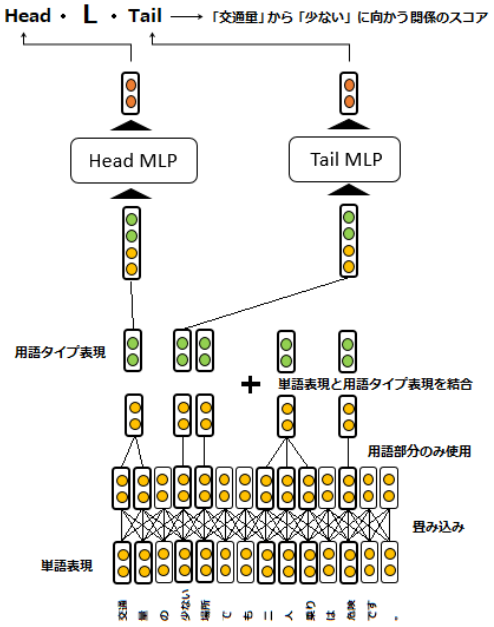


図 1: 作成したモデルの概要図

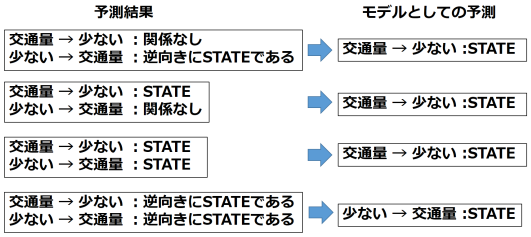


図 2: 判断方法

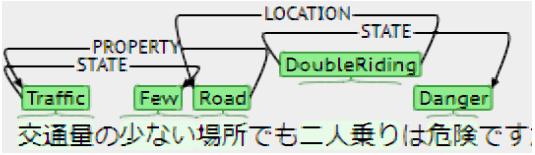


図 3: タグ付けされた文の例

# 4 実験

## 4.1 実験設定

本研究では先行研究 [1] で使用された，MeCab による形態素解析と用語のタグ付けが行われた交通文書とアノテーションファイルを用いた．データについて表 2 に示す．ただし，文の中には用語を含まないものや，1 つだけしか用語を含まず文中に関係が存在し得ないものもある．そのような文は前処理の段階で省き学習を行った．用語のペアは 1 文中に存在する用語のすべての組み合わせを考える．そのため 1 文中の用語数が $n$ 個である場合用語ペア数は ${}_nC_r$ となる．

図 3 に文の例を示す．例にある「STATE」,「PROPERTY」,「LOCATION」はそれぞれ状態，特性，位置を表す関係である．このような関係が 66 種類タグ付けされている．

この 66 種類の関係それぞれについて先行研究 [1] と同様に 5 分割交差検証による精度を F 値として求めた．

F 値について，関係があるペアを「関係がある」と予測した場合 True Positive (TP)，関係がないペアを「関係がある」と予測した場合 False Positive (FP)，関係があるペアを「関係がない」と予測した場合 False Negative (FN)，関係がないペアを「関係がない」と予測した場合 True Negative (TN) とする．適合率 (Precision)，再現率 (Recall) は以下の式で計算される．

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

表 2: 交通文書データの統計

| | 数 |
|---|---|
| 文数 (全体) | 1,984 |
| 文数 (用語を 2 つ以上含む) | 1,207 |
| 用語ペア数 | 30,410 |
| 関係数 | 66 |
| 関係のあるペア数 | 6,278 |

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

これらを用いて F 値は以下のように計算する．

$$F = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{6}$$

F 値は 0 から 1 の間で与えられ，1 に近づくほど高精度であることを示す．

単語埋め込みの初期値として，fastText モデルによって Wikipedia で学習した事前学習済みベクトル[1]を用いる．また，モデルを学習する際に使用したパラメータを表 3 に示す．このハイパーパラメータは，予備実験において一部の関係のみを対象にしてチューニングを行ったものであり，このチューニングは今後の課題である．

## 4.2 結果と考察

表 4 に結果を示す．ほぼすべての関係において先行研究と比べて高い精度が得られた．交通用語間の関係抽出において，SVM よりも CNN が有用であることがわかった．

また，全関係における F 値 (micro F 値) は 0.408 であった．F 値が 0 となった関係は多く見られるが，それに比べて micro F 値は高い．これは正例が少ないため関係を学習できなかった関係が多いためである．関係によっては正例が数個しかないため学習部分またはテスト部分にかたまり，学習，テストに使用できる正例が 0 である場合が発生した．このような頻度の少ない関係に対処するために，データの分割を考える必要

表 3: 学習設定

| | |
|---|---|
| 使用言語 | Python 3.6.5 |
| 使用ライブラリ | PyTorch 0.4.1 |
| optimizer | Adam |
| バッチサイズ | 32 |
| 単語表現の次元 | 300 |
| 用語タイプ表現の次元 | 60 |
| 隠れ層の次元 | 150 |
| 学習率 | 0.001 |
| エポック数 | 50 |
| フィルタサイズ | 3,5,7 |

[1] https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md

があるとともに，正例の追加や似た関係の統合などが必要であると考えられる．

## 5 おわりに

本研究では CNN が交通用語間の関係抽出において有用であるか検証を行い，CNN を用いることで，SVM を用いた場合に比べて，精度が向上することがわかった．しかし，全体の micro F 値で 0.408 であり，改善の必要がある．また，正例が少ないためにうまく学習できなかったと考えられる関係が多く見られた．

今後の課題としては，CNN のフィルタサイズやフィルタの数，畳み込みの層の数についての調整は不十分であり，このような部分の調整により CNN を用いたモデルの性能をより詳細に調査する必要がある．また，正例が少ない関係について，アノテーションデータの追加や関係の種類の統合により，学習例の数を増やす必要がある．

表 4: 関係抽出精度 (F 値)

| 関係種類 | 先行研究 | 本研究 |
|---|---|---|
| ACT | 0.103 | 0.316 |
| AFTERTIME | 0.043 | 0.339 |
| AMOUNT | 0.000 | 0.108 |
| APPRISE | 0.054 | 0.118 |
| ATSPEED | 0.045 | 0.364 |
| AVOID | 0.000 | 0.167 |
| BEFORETIME | 0.023 | 0.199 |
| BELONG | 0.000 | 0.000 |
| BETWEEN | 0.000 | 0.000 |
| CAUSE | 0.103 | 0.118 |
| CONDITION | 0.069 | 0.337 |
| CONTENT | 0.089 | 0.518 |
| CONTRADICT | 0.037 | 0.000 |
| COORDINATION | 0.277 | 0.442 |
| DECISION | 0.000 | 0.000 |
| DECREASE | 0.000 | 0.128 |
| DESTINATION | 0.009 | 0.483 |
| DIRECTION | 0.203 | 0.499 |
| DISTANCE | 0.031 | 0.473 |
| DRIVE | 0.214 | 0.603 |
| EXCEPT | 0.064 | 0.296 |
| EQUIVALENCE | 0.009 | 0.202 |
| INCREASE | 0.000 | 0.000 |
| LENGTH | 0.000 | 0.000 |
| LOCATION | 0.194 | 0.427 |
| OBEY | 0.003 | 0.273 |
| OVERAGE | 0.000 | 0.013 |
| OVERAMOUNT | 0.000 | 0.000 |
| OVERDISTANCE | 0.000 | 0.000 |
| OVERLENGTH | 0.000 | 0.000 |
| OVERHEIGHT | 0.000 | 0.000 |
| OVERPEOPLE | 0.000 | 0.000 |
| OVERSPEED | 0.000 | 0.177 |
| OVERTIME | 0.000 | 0.000 |
| OVERWEIGHT | 0.000 | 0.000 |
| OVERYEAR | 0.000 | 0.100 |
| PART | 0.067 | 0.451 |
| PAY | 0.000 | 0.000 |
| PERMIT | 0.038 | 0.390 |
| POSITION | 0.072 | 0.250 |
| PROPARTY | 0.065 | 0.442 |
| PURPOSE | 0.084 | 0.281 |
| RATIO | 0.000 | 0.000 |
| RELATIVEPOSITION | 0.114 | 0.508 |
| REQUIRE | 0.098 | 0.336 |
| SOURCE | 0.071 | 0.290 |
| SPECIFY | 0.122 | 0.272 |
| STATE | 0.292 | 0.456 |
| SUBCONCEPT | 0.076 | 0.267 |
| TARGET | 0.140 | 0.424 |
| TIME | 0.099 | 0.280 |
| TIMES | 0.024 | 0.054 |
| TOOL | 0.113 | 0.482 |
| UNCONDITON | 0.000 | 0.000 |
| UNDERAGE | 0.000 | 0.267 |
| UNDERCAPACITY | 0.000 | 0.000 |
| UNDERDISTANCE | 0.041 | 0.726 |
| UNDERHEIGHT | 0.000 | 0.000 |
| UNDERLENGTH | 0.034 | 0.303 |
| UNDERPOWER | 0.000 | 0.000 |
| UNDERTIME | 0.000 | 0.000 |
| UNDERSPEED | 0.032 | 0.242 |
| UNDERWEIGHT | 0.000 | 0.000 |
| UNDERYEAR | 0.000 | 0.073 |
| USE | 0.068 | 0.541 |
| UNEQUIVALENCE | 0.064 | 0.493 |

## 参考文献


[1] 河辺一仁．交通オントロジーの半自動構築のための用語・関係抽出．豊田工業大学院修士論文, 2016.

[2] D Zeng, et al. Relation classification via convolutional deep neural network daojian. Proceedings of COLING 2014, 2014.

[3] 国家公安委員会．交通の方法に関する教則．全日本交通安全協会, 2012.

[4] 工藤拓, 松本裕治．チャンキングの段階適用による日本語係り受け解析．Vol. 43, No. 6, pp. 1834–1842, 2002.

[5] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, Vol. 5, pp. 135–146, 2017.

[6] Patrick Verga, Emma Strubell, and Andrew McCallum. Simultaneously self-attending to all mentions for full-abstract biological relation extraction. pp. 872–884. Association for Computational Linguistics, 2018.