論理関係を自然言語から抽出する研究

Research on Logical Relationships Extraction from Natural Language

王 梓萱 1)

指導教員　亀田　弘之 2)

1) 東京工科大学　バイオ・情報メディア研究科　コンピュータサイエンス専攻

2) 東京工科大学　コンピュータサイエンス学部　先進情報専攻

キーワード：関係抽出、ニューラルネットワーク、依存ツリー、自己注意

1．研究背景と目標

関係抽出タスクは、自然言語処理の代表的なタスクの 1 つであり、非構造化テキストからエンティティとエン ティティ間の意味関係を確定することを目的としている。関係抽出により、構造化されていないテキストから効果的な情報を自動的に取得してナレッジを形成することができる。

2012 年にナレッジグラフが提案された後、質問応答システムや検索エンジンなどのアプリケーションで多くの大規模なナレッジグラフが継続的に使用され、現代の社会や生活におけるナレッジグラフの重要性が示されている。

しかし、既存のナレッジグラフのナレッジは、現実の世界では非常に少ない量のナレッジしか占めていない。ナレッジグラフを作成するために人的資源に依存することは、莫大な経費と時間が必要である。そのため、多くの研究者が、関係抽出の研究をしている。

本研究は、文の依存関係ツリーを最大限に活用するために、文依存ツリーの自己注意に基づく関係抽出(Relationship extraction based on self-attention of tree、RESAT)という手法を提案し、その妥当性の検証を行う。

2. 依存ツリー自己注意に基づく関係抽出の提案

2.1 依存ツリー自己注意に基づく関係抽出

文依存ツリーの自己注意に基づく関係抽出の構造を図 1 に示す。LSTM、自己注意、線形マージ、プーリング、フィードフォワードニューラルネットワークから構成されている。

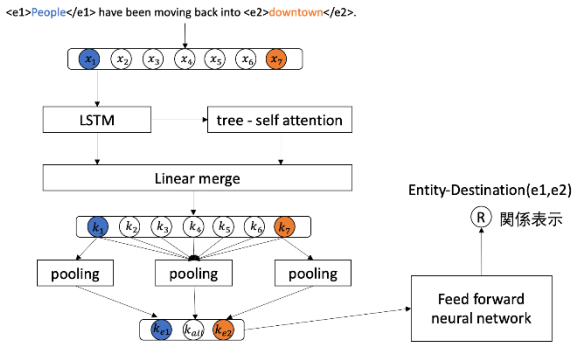

図 1　依存ツリー自己注意に基づく関係抽出の構造

2.2 依存ツリー自己注意に基づく関係抽出

長さ n の文 $T = [t_1, t_2, \ldots, t_n]$ を想定し、最初に GloVe 単語ベクトルライブラリを使用して単語のベクトルを生成し、次に LSTM 処理を使用して文

のコンテキストを持つベクトル$L = [l_1, l_2, \dots, l_n]$を生成する。

3 つの線形変換により、i番目のベクトルのq_i, k_i, v_iを取得する。文献[1]の用語$\left(Query(q), Key(k), Value(v)\right)$を使用して以下概要を紹介する.

$$\begin{aligned} q_i &= W_q l_i \\ k_i &= W_k l_i \\ v_i &= W_v l_i \end{aligned} \tag{3.1}$$

ここで、W_q, W_k, W_vは重み行列である。

i番目のベクトルに対するj番目のベクトルの重要性は、i番目のベクトルのq_iとj番目のベクトルのk_jを使用する関数fで計算する（dはベクトル次元)。

$$f(q_i, k_j) = \frac{q_i k_j}{\sqrt{d}} \tag{3.2}$$

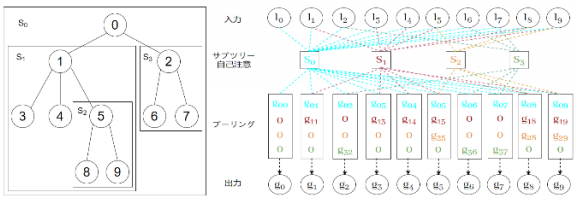

図 2　依存ツリー自己注意の回路図

依存関係ツリーの情報を最大限に活用するために、サブツリーで自己注意メカニズムを使用する。具体的には、サブツリーsのi番目のベクトルに対するj番目のベクトルの重要性は

$$a_{ij}^{s} = \begin{cases} \dfrac{exp\left(f(q_i, k_j)\right)}{\sum_{p \in s} exp\left(f(q_i, k_j)\right)}, i, j \in s \\ 0, otherwise \end{cases} \tag{3.3}$$

なので、$v_1, v_2, \dots, v_n$の加重平均合計を計算し、サブツリーsのi番目のベクトルを次のように更新する。

$$g_{si} = \sum_{j \in s} a_{ij}^{s} v_j \tag{3.4}$$

サブツリーでの自己注意（式 3.3 と式 3.4）により、各サブツリー上の各ノードの更新ベクトルを取得できる。しかし、図 2 によりノードは同時に複数のサブツリーに属することが分かる。異なるサブツリーには異なるノードが含まれているため、ノードは異なるサブツリー毎に異なる更新ベクトルを取得する。i番目のノードを含むすべてのサブツリーをセット$S(i)$として扱い、$S(i)$のi番目のノードの更新ベクトルもセットを構成する。

$$\{g_{si} | s \in S(i)\} (i \in s, s \in S(i)) \tag{3.5}$$

最後に、i番目のノードのすべての更新ベクトル$\{g_{si} | s \in S(i)\}$をプーリングし、依存ツリーの自己注意でi番目のノードの出力を取得する。

$$g_i = pooling(\{g_{si} | s \in S(i)\}) \tag{3.6}$$

3. 考察

3.1　データセット

SemEval-2010 Task8 のデータセットには 19 の関係が含まれ、そのうちの 1 つは人工的な関係 other と 9 組の方向がある関係（18 の関係）である。10217 個のサンプルが含まれ、トレーニングセットには 8000 個、テストセットには 2717 個のサンプルが含まれている。

3.2　評価指標

SemEval の公式評価指標はマクロ平均 F1(macro-averaging F1)

$$macro - averagingF1 = \frac{\sum_{c=1}^{C} F1_c}{C} \tag{4.1}$$

ここで、Cは関係の数、$F1_c$はc番目の関係の$F1$である。

3.3 実験

GloVe の単語ベクトルを使用し、LSTM をライヤー数は 1、隠れ層の次元は 300 に設定して構築した。依存ツリーの自己注意モジュールをプログラミングしたら、精度向上の程度を検証してみる。

4. 結論

本研究は、関係抽出を目指して文の依存関係ツリーで最大限に活用を行っており、現在までに数学的に関連研究と比べて SemEval の関係抽出に対する精度向上が有望であると考えられる。

参考文献

[1] VASWANI A,SHAZEER N,PARMAR N,et al. Attentions is all you need [C/OL] Proc. of NeurIPS. 2017.