4K1-5

Recursive AutoEncoderを用いた文間の接続関係推定

大塚 淳史 Atsushi OTSUKA
平野 徹 Toru HIRANO
宮崎 千明 Chiaki MIYAZAKI
東中 竜一郎 Ryuichiro HIGASHINAKA
牧野 俊朗 Toshiro MAKINO
松尾 義博 Yoshihiro MATSUO

日本電信電話株式会社 NTT メディアインテリジェンス研究所
NTT Media Intelligence Laboratories, NTT Corporation

Relation recognition between sentences is important for many applications such as dialog systems or text summarization. We present a method for relation recognition using concept vectors of words, segments, predicate argument structures and sentences. Concept vectors are created by a Recursive AutoEncoder which is a kind of Neural Networks. Experimental results show the proposed method outperformed previous methods that uses bag-of-words or centroids of concept vector.

1. はじめに

因果関係や対比関係など，二文間の論理的な繋がりを推定する接続関係推定は対話や要約，文生成など様々なタスクへの応用が期待できるため，自然言語処理分野における重要な課題の一つとなっている [山本 08]．例えば，ユーザ発話に対して，大量の発話候補から発話を選択する対話システムでは，ユーザ発話に対して不適切な返答をしてしまうという問題が存在する [東中 15]．発話間の接続関係が明らかとなればユーザの発話に対して論理的な繋がりある返答ができるようになる．接続関係を持つ 2 文は首尾一貫性が保たれるため，接続関係を考慮して発話を行うことで対話システムの発話の性能が向上することが期待できる．

接続関係は，接続詞や接続助詞などの接続表現が明示的に示されている Explicit 接続関係と，接続表現は明記されていないが二文間の意味の関係性から接続関係が読み取れる Implicit 接続関係の二種類に分類できる [Prasad 08]．Explicit 接続関係については接続表現を素性とした手法で人手と同程度の精度で推定できる [Pitler 09]．しかし，Implicit 接続関係については Explicit 接続関係と比べ推定精度が低く，改善の余地がある [Lan 13]．

Implicit 接続関係を推定するには，二文間の内容を比較する必要がある．従来研究では，二文に出現する単語のペアを素性とすることで二文間の内容の比較を行っている [Lin 09] が，単語の比較だけでは正しく接続関係を推定できない問題がある．例えば，“僕は夏が好き” と “私は冬のほうがいい” という 2 文については “夏” と “冬” という単語を比較することで対比関係であることがわかる．しかし，同じ対比関係であっても “久々に山に登りたいな” と “私はインドア派なんで” の二文では “山に登る” と “インドア” という述語項と単語を比較しなければ，対比関係であることがわからない．このように二文間の接続関係を推定する際には，二文間の内容の比較を単語，文節，述語項あるいは文全体など，様々な粒度で行う必要がある．

本論文では，Recursive AutoEncoder（RAE）を用いた概念ベクトルによる文間の接続関係推定手法を提案する．RAE は，任意の単語長の文に対して，ニューラルネットワークを用いて再帰的にベクトル合成を繰り返すことで，文節や文の概念ベクトルを作成する手法である [Socher 11]．RAE により作成された概念ベクトルは入力した単語の概念ベクトルと同一の次元数となるため，単語と文節，文節と述語項，文節と文など文内の様々な粒度での意味の比較を行うことが可能になる．また，RAE は文の構造情報を考慮した概念ベクトルを生成することができるため，単なる内容語の重心とは異なり，各接続関係に特有の言い回しなども捉えることができる．

Explicit 接続関係を持つ二文は接続詞や接続助詞を手掛かりとして容易に収集することができる．そこで本論文では，事前に収集した Explicit 接続関係の二文を教師データとして収集し学習した接続関係推定器を，Implicit 接続関係の二文に適用し，提案手法の有効性を検証する．

連絡先: 大塚淳史，NTT メディアインテリジェンス研究所，神奈川県横須賀市光の丘 1-1，otsuka.atsushi@lab.ntt.co.jp

2. 関連研究

接続関係は，二文間の関係性の中でも特に，「でも」，「だから」，「例えば」の他，「から」や「とか」などの接続表現によって表される関係である．近年では，The Wall Street Journal の新聞記事に接続関係のタグ付けを行った Penn Discourse Tree Bank(PDTB)[Prasad 08] コーパスを利用した，接続関係推定について様々な研究がなされている．

Lin ら [Lin 09] は PDTB の接続関係タグセットを用いて，単語の共起ペア，文構造といった表層表現による接続関係推定器を提案している．また，Lan ら [Lan 13] は，Implicit 接続関係推定を接続詞を決定する問題と，接続詞から接続関係を決定する問題という二つのタスクに分割し，マルチタスク学習で学習する手法を提案している．

日本語における接続関係推定については，山本ら [山本 08] の研究がある．山本らは用例検索の概念を接続関係推定に導入し，接続関係が既知の二文と接続関係が未知の二文を構文，単語の類似性から比較し，最も類似度が高い接続関係を割り当てるという手法を提案している．

近年では，単語や文間の関係推定に Deep Learning と呼ばれるニューラルネットワークを用いる手法が多く提案されている．Zeng ら [Zeng 14] は，文中に出現する単語間の関係性を Deep Learning により推定する手法を提案している．Socher ら [Socher 11] は RAE によって生成した概念ベクトルを用いて二文間の言い換え判定タスクに取り組み，従来の SVM を用いた手法よりも高精度に言い換え判定が行えることを明らかにしている．また，Li ら [Li 14] は，修辞構造理論に基づく．文

書の談話構造の推定に RAE を適用する手法を提案している．

本論文で提案する手法は，RAE により，文の意味を抽象化した概念ベクトルを用いて接続関係推定を行う点で従来手法と異なる．概念ベクトル上では意味が近い単語は近い座標に分布されるため，教師データに含まれない単語が使用されている文に対しても，概念ベクトルの距離を比較することにより対応できるという利点がある．また，RAE は文の構文情報に従い，ベクトル合成を行うため RAE によって作成された概念ベクトルは構文も含めた文の意味を捉えている．単語の類似性と構文情報により推定する手法は山本ら [山本 08] の推定手法に近いが，山本らは構文情報をデータベースに保存していなくてはいけないのに対し，提案手法では構文情報を概念ベクトルに内包させるため，構文情報を蓄積する必要がない点が異なる．

3. 提案手法

本節では，Recursive AutoEncoder(RAE) を用いた接続関係推定手法について説明する．まず，3.1 で RAE を用いた概念ベクトルの作成手法について説明し，次に，3.2 で概念ベクトルによる接続関係推定手法について説明する．

3.1 Recursive AutoEncoder

RAE は Socher ら [Socher 11] が提案した，文節や文の概念ベクトルの作成手法である．RAE では，ニューラルネットワークを用いて，単語ベクトルを再帰的に繰り返し合成することでボトムアップ的に文節や文の概念ベクトルを作成する．RAE モデルの例を図 1 に示す．N 次元の概念ベクトル c_1，c_2，c_3 を持つ 3 単語から構成される文があるとき，RAE ではまずベクトル c_1 と c_2 の合成を行う．c_1 と c_2 を連結し $2N$ 次元の $[c_1; c_2]$ というベクトルを作成する．合成ベクトル p_1 は，符号化重み行列 W_e とバイアス項 b_e，活性化関数 f からなるニューラルネットワークにより，以下の通り計算できる．

$$p_1 = f(W_e[c_1; c_2] + b_e) \tag{1}$$

ここで，符号化重み行列を $N \times 2N$ の行列とすると，合成ベクトル p_1 は N 次元となる．この計算により，合成ベクトルと単語ベクトルが同一の次元で表現できるようになる．合成ベクトル p_1 と残りの単語ベクトル c_3 を同様に合成することで，文ベクトル p_2 が作成される．

符号化重み行列 W_e とバイアス項 b_e は教師なし学習によりパラメータ学習を行う．合成ベクトル p_2 に対して，合成と逆順で，復号化重みパラメータ W_d, b_d によりベクトルを展開していく．最終的に入力ベクトル c_1，c_2，c_3 に対して，再現ベクトル c_1'，c_2'，c_3' が生成される．ここからこの RAE 合成ベクトルの再現誤差 $Err_{rep}(p_2)$ は以下の式で計算できる．

$$Err_{rep}(p_2) = ||[c_1; c_2; c_3] - [c_1'; c_2'; c_3']||^2 \tag{2}$$

これを各合成ベクトルで計算し，その合計が RAE 全体の再現誤差となる．以降は再現誤差が小さくなるように勾配法などでパラメータを学習していく．

3.1.1 Recursive AutoEncoder の日本語への適用

RAE ではベクトル合成を構文情報に従って合成していく．英語の構文木は句構造規則に基づく 2 分木により表現されているため，構文木に従い合成を行えばよいが，日本語の構文は基本的には係り受け構造となるという違いがある．係り受け構造では一つの文節に複数の文節が係るため，単純な 2 分木とはならない．そこで本論文では，日本語の係り受け構造に基づき，単語から文節，述語項，文の順にベクトルの合成を行う．

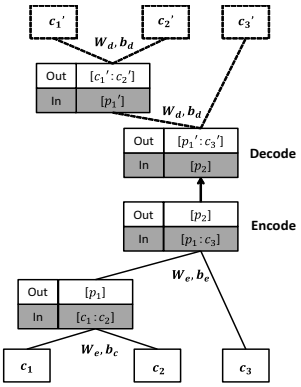

図 1: Recursive AutoEncoder のモデル

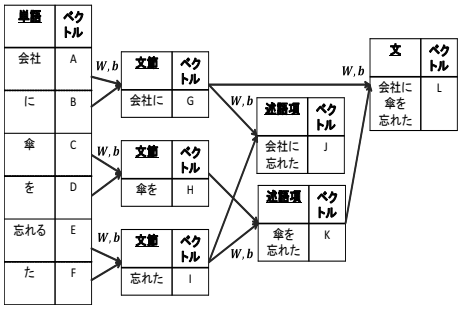

図 2: RAE による日本語ベクトルモデル

日本語の文に RAE を適用した例を図 2 に示す．まず，単語ベクトルを合成し文節ベクトルを作成する．単語ベクトルを先頭から順に合成していくことで文節ベクトルを作成する．次に，文節ベクトル同士を合成し，述語項ベクトルを作成する．日本語の文に係り受け解析を実行し，述部に対応する文節を決定する．述部に対応するベクトルとその文節に係る文節を一つ選択し，RAE によりベクトル合成することで述語項ベクトルを作成する．このとき，述語項ベクトルは述部に係る全てのベクトルで作成する．図 2 の例では，述部となる文節 “忘れた” に対して “会社に” と “傘を” の 2 つの文節が係るので，それぞれでベクトル合成を行い，“会社に忘れた” と “傘を忘れた” という 2 つの述語項ベクトルを作成する．

文ベクトルは係り受け構造に従い，述部となる文節に近い文節から順に合成を行う．図 2 の例では，述部となる文節 “忘れた” に近い位置にある文節 “傘を” と合成し，“傘を忘れた” という述語項ベクトルを作成する．次に残りの文節 “会社に” と述語項ベクトル “傘を忘れた” を合成し，“会社に傘を忘れた” という文ベクトルを作成する．ここで，述部と合成する文節ベクトルが他の文節の係り先となっている場合はまず，これらの文節同士でベクトル合成を行った後，その合成ベクトルを述部の文節ベクトルと合成する．

3.2 概念ベクトルを用いた接続関係推定

文間の接続関係推定のため, RAE により作成した概念ベクトルを用い, 文間の意味の類似度，構文を含む文の意味情報，概念ベクトル上での位置関係の 3 点に着目して素性を作成する．以降はそれぞれについて詳述する．

3.2.1 Dynamic Pooling による 2 文間の類似度

Dynamic Pooling は，Socher ら [Socher 11] が提案した概念ベクトル上での 2 文間の類似性を比較する手法である．Dynamic Pooling により，単語と文節，文節と述語項，単語と文

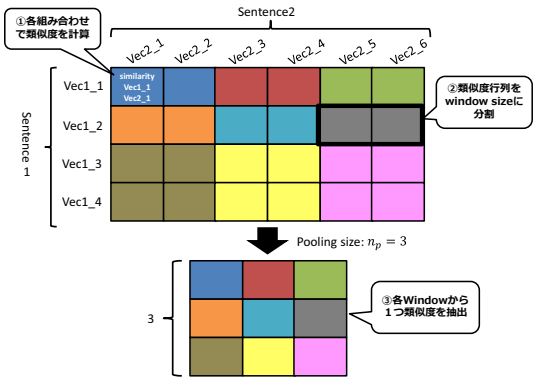

図 3: Dynamic Pooling の概要

など異なる粒度の表現間の類似度を比較し，固定長の次元に落としこむことが可能になる．Dynamic Pooling の概要を図 3 に示す．Dynamic Pooling では，RAE で作成した 2 文各々の単語ベクトル，文節ベクトル，述語項ベクトル，文ベクトルの類似度を全組み合わせで計算し，類似度行列を作成する．類似度行列は入力文の単語長に依存して変化するため，接続関係推定の素性として利用するには類似度行列のサイズの固定する必要がある．Dynamic Pooling では設定した素性行列のサイズに応じて，類似度行列の部分行列を抽出し，その部分行列で最も類似度が高い要素を一つ選択する．図 3 では，部分行列を色分けで表現している．N 個のベクトルを持つ文 1 と，M 個のベクトルを持つ文 2 を与え，固定長の素性行列のサイズを n_p と設定した時，部分行列のサイズは $[N/n_p] \times [M/n_p]$ となる．式が割り切れない場合は，最後の部分行列のサイズを変更して対応する．図 3 では文 1 の最後で部分行列のサイズを 1×2 から 2×2 に変更している．

3.2.2 構文情報を含む意味の概念ベクトル

RAE では，文の構文に従いベクトルの合成を行っていくため，RAE によって作成された概念ベクトルは構文の情報を含むことになる．似た意味の単語を使用し，かつ，似た構文で記述された文は概念空間上で近い位置に分布する．つまり，同じ単語が使われているという情報だけでなく，“どのような言い回し” をしたかという情報が含まれている．従って，RAE で作成した文ベクトルを推定に用いることで文の語順や構文情報を考慮することが可能となる．これは，単語ベクトルの重心による文ベクトルでは考慮することのできない情報である．

RAE と単語ベクトルの重心により得た文ベクトルを主成分分析により 2 次元に可視化した結果を図 4 に示す．左が単語ベクトルの重心によって得られた文ベクトルの分布，そして右が RAE によって得られた文ベクトルの分布である．RAE では “野球よりもサッカーが好きです” と “サッカーよりも野球が好きです” の文ベクトルが離れたところに分布している．一方，単語ベクトルの重心では，この 2 文は完全に同じ場所に分布している．これは，単語ベクトルの重心によって得られた文ベクトルでは語順や構造が考慮できないためである．接続関係推定ではこれらの違いを考える必要がある．例えば，1 文目が “野球が好きです”，2 文目が “サッカーよりも野球が好きです” とした場合は，“野球が好き” であることに付加情報を追加している関係性にあたる．一方で，2 文目が “野球よりもサッカーが好きです” の場合は “野球が好き（だけど）野球よりもサッカーが好き” という逆説的なの関係になる．単語ベクトルの重心から求めた文ベクトルでは，これらの違いに対応できないことになる．

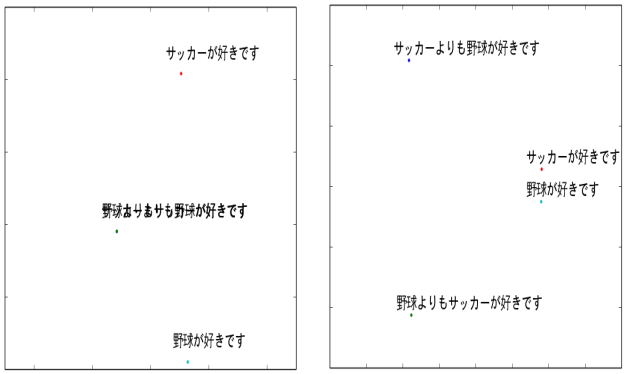

図 4: 文ベクトル分布の比較 (左) 単語ベクトル重心 (右)Recursive AutoEncoder によるベクトル合成

表 1: 接続関係と対応する接続表現

接続関係	接続表現
因果	だから，ので，なので，だったら，じゃあ
比較	でも，しかし，けど，だけど，だが
展開	あと，例えば，しかも，さらに，つまり

3.2.3 差分ベクトルによる意味の関係性

接続関係推定は 2 文間の意味関係を推定するものである．概念ベクトルは，意味が近い単語や文が概念空間上で近い位置に分布する．言い換え推定などの二文間の意味の類似性を比較するのであれば，3.2.1 の Dynamic Pooling の様に二文の類似度を比較すればよい．しかし，接続関係推定で比較する二文は必ずしも意味が近いとは限らず，単純な類似度だけでは，比較できない．

本論文では，二文の意味の関係性を概念空間上のベクトルの位置関係に対応させて，二文のベクトルの差分ベクトルを使用する．差分ベクトルは，概念空間上の二点間の相対的な関係性を表現することができるため，教師データと意味が類似していない二文に対しても，その位置関係から接続関係が推定できることが期待される．

素性とする差分ベクトルは文 1，文 2 の各単語，文節，述語項ベクトル全ての組み合わせで差分ベクトルを計算し、重心を計算したものを使用する．

4. 評価実験

本節では，対話における発話文を対象に接続関係推定器の評価を行う．実験のデータセットとして，東中ら [Higashinaka 14] の雑談対話コーパスを用いる．コーパスから PDTB のタグセットを参考に，“因果（CONTINGENCY）”，“比較 (COMPARISION)”，“展開（EXPANSION）” の 3 種類の接続関係を持つ発話対を，表 1 の接続表現を手掛かりに 15,000 対収集した．テストセットとして接続表現を含まないが，人手で接続関係があると判断された発話対を各接続関係で 500 対収集した．

接続関係推定の学習器には SVM を用いる．以下の素性により学習した学習器について，精度・再現率・F 値を比較する．単語ベクトルは日本語版 Wikipedia コーパスに Word2Vec を適用して作成したものを用いる．また RAE のパラメータは雑談対話コーパスの全発話を適用して学習している．

手法 A bag-of-words
手法 B 単語ベクトル重心による文ベクトル
手法 C RAE による文ベクトル

表 2: 発話対の Implicit 接続関係推定実験結果

	因果			比較			展開		
	精度	再現率	F 値	精度	再現率	F 値	精度	再現率	F 値
手法 A（bag-of-words)	0.38	0.64	0.47	0.40	0.29	0.33	0.45	0.22	0.29
手法 B（単語ベクトル重心）	0.45	0.32	0.37	0.43	0.43	0.43	0.47	0.40	0.43
手法 C（RAE 文ベクトル）	0.42	**0.40**	**0.41**	**0.50**	0.26	0.34	0.39	**0.60**	**0.47**
手法 D (RAE + 差分ベクトル）	0.42	**0.36**	**0.39**	0.46	0.26	0.33	0.39	**0.61**	**0.48**
手法 E (RAE + Dynamic pooling)	0.41	**0.41**	**0.41**	0.46	0.29	0.36	0.39	**0.53**	**0.45**
手法 F (RAE + Dynamic pooling + 差分ベクトル)	0.45	**0.41**	**0.43**	**0.48**	0.30	0.37	0.41	**0.60**	**0.49**

表 3: 手法 F での接続関係推定結果例

発話対		接続表現
発話文 1	発話文 2	
この時間の料理の話しはヤバいですね	めちゃ腹へってきました	因果
私もスキー得意です！	ボードは難しくて	比較
好きな番組のジャンルありますか	バラエティですかね	展開

手法 D RAE + 差分ベクトル
手法 E RAE + Dynamic pooling(size:5)
手法 F RAE + Dynamic pooling + 差分ベクトル

推定結果を表 2 に示す．提案手法である手法 C〜F について，比較手法である手法 A，B よりもスコアが上回っているものを太字，また提案手法内の各評価でスコアが最も高かったものを下線で示している．また，手法 F により接続関係を推定できた発話対を表 3 に示す．

比較関係では，提案手法である手法 C〜F が，ベースラインである重心法の手法 B よりも F 値で下回る結果となった．比較関係は，2 つの対象を比較する関係であるが，比較する対象は表 3 の例の様に “スキー” と “ボード” の様に単語間で比較しているものが多い．Word2Vec の単語ベクトルをそのまま使用している手法 B では，単語間の関係が汎化され分類精度が向上したと考えられる．一方，RAE を使った手法では，単語だけでなく文節や述語項，文の比較も同時に行っており，結果的に単語間の比較の情報が落ちてしまっているため，比較での再現率が低下したのではないかと考えられる．

因果関係と展開関係では，手法 B と比べ，提案手法が F 値で上回っている．因果関係と展開関係は比較関係よりも “言い回し” の表現が多様である．そのため，構文情報を考慮できる RAE を適用した手法が，重心法による手法よりも優位な結果になったと考えられる．また，提案手法の中では，RAE の文ベクトルだけでなく，Dynamic Pooling と差分ベクトルを組み合わせた手法 F が最も良い結果となっており，概念ベクトルの空間上でそれぞれが効果的に機能していると考えられる．

5. おわりに

本論文では，RAE を用いて，構文情報を考慮しながら文節，述語項，文など異なる粒度の表現を概念ベクトル化することによる，2 文間の接続関係推定手法を提案した．ベクトルの類似性を比較する Dynamic Pooling，構文情報を含む文ベクトル，そして 2 文の差分ベクトルを用いることで，接続表現が含まれない 2 文についても，従来手法より高精度で接続関係を推定できることを示した．

今後は，概念ベクトルの構築手法，接続関係に適した素性表現についての検討を進めるほか，コンテキストなど，概念ベクトル以外の素性との組み合わせによる接続関係推定についても取り組んでいきたいと考えている．

参考文献

[Higashinaka 14] Higashinaka, R., Imamura, K., Meguro, T., Miyazaki, C., Kobayashi, N., Sugiyama, H., Hirano, T., Makino, T., and Matsuo, Y.: Towards an Open Domain Conversational System Fully Based on Natural Language Processing, *Proc of the 25th International Conference on Computational Linguistics (COLING 2014)*, pp. 928–939 (2014)

[Lan 13] Lan, M., Xu, Y., and Niu, Z.: Leveraging Synthetic Discourse Data via Multi-task Learning for Implicit Discourse Relation Recognition, *Proc of the 51st Annual Meeting of the Association for Computational Linguistics(ACL 2013)*, pp. 476–485 (2013)

[Li 14] Li, J., Li, R., and Hovy, E.: Recursive Deep Models for Discourse Parsing, *Proc of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pp. 2061–2069 (2014)

[Lin 09] Lin, Z., Kan, M.-Y., and Ng, H. T.: Recognizing Implicit Discourse Relations in the Penn Discourse Treebank, *Proc of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009)*, pp. 343–351 (2009)

[Pitler 09] Pitler, E. and Nenkova, A.: Using Syntax to Disambiguate Explicit Discourse Connectives in Text, *Proc of the ACL-IJCNLP 2009 Conference*, pp. 13–16 (2009)

[Prasad 08] Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A., and Webber, B.: The Penn Discourse TreeBank 2.0, *Proc of the sixth international conference on Language Resources and Evaluation (LREC 2008)* (2008)

[Socher 11] Socher, R., Huang, E. H., Pennin, J., Manning, C. D., and Ng, A. Y.: Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection, *Proc of Advances in Neural Information Processing Systems (NIPS 2011)*, pp. 801–809 (2011)

[Zeng 14] Zeng, D., Liu, K., Lai, S., Zhou, G., and Zhao, J.: Relation Classification via Convolutional Deep Neural Network, *Proc of the 25th International Conference on Computational Linguistics(COLING 2014)*, pp. 2335–2344 (2014)

[山本 08] 山本 和英, 齋藤 真実 : 用例利用型による文間接続関係の同定, 自然言語処理, Vol. 15, No. 3, pp. 21–51 (2008)

[東中 15] 東中 竜一郎, 船越 孝太郎, 荒木 雅弘, 塚原 裕史, 小林 優佳, 水上 雅博 : Project Next NLP 対話タスク : 雑談対話データの収集と対話破綻アノテーションおよびその類型化, 言語処理学会第２１回年次大会ワークショップ　自然言語処理におけるエラー分析論文集 (2015)