Holographic Embeddings による CCG 構文解析

山木良輔 [1]　谷口忠大 [1]　持橋大地 [2]
[1] 立命館大学 情報理工学部　[2] 統計数理研究所
{yamaki.ryosuke,taniguchi}@em.ci.ritsumei.ac.jp　daichi@ism.ac.jp

概要

本研究では知識グラフをベクトル空間に埋め込むための手法である Holographic Embeddings を CCG 構文解析に適用することで，単語分散表現から句や文の分散表現を再帰的に構成し，文の構成要素間の依存関係を明示的にモデリングする手法を提案する．そして，提案手法が CCG における Supertagging 精度の向上に有効であることを示し，さらに句の分散表現を用いることで句に対するカテゴリ割当を直接的に予測しながら句構造を探索する Span-based Parsing アルゴリズムを導入する．実験結果から，提案手法は Supertagging 精度において既存モデルを上回る性能を示し，構文解析の精度においても，同じ C&C パーザを用いる中で世界最高性能を達成した．

1　はじめに

自然言語処理において，文の背後に潜む統語構造を解析する技術として構文解析が存在する．構文解析は統語的な曖昧性の解消に役立つため，質問応答や機械翻訳など，様々な自然言語処理アプリケーションの基盤技術として活用されている．構文解析のうち，特に単語間に存在する階層関係・句構造を明らかにすることに焦点をおいたものとして句構造解析，そして解析に用いられる文法理論の１つとして組み合わせ範疇文法 (Combinatory Categorial Grammar, CCG) [1] がある．CCG は語彙化文法の一種であり，文中の各単語に対してその単語の文法的機能を示すカテゴリを与え，それらを組み合わせ規則に従って組み合わせることで統語構造を計算する．ここで，各カテゴリは S: 文，N: 名詞，NP：名詞句などの基本カテゴリと"/"，"\" を再帰的に組み合わせることによって構成され，単語の文法的機能に関する情報を多く含む．CCG を用いることで単語間の長距離依存関係や並列構造など，自然言語に存在する複雑な統語構造を簡潔に表現可能であることが知られている．

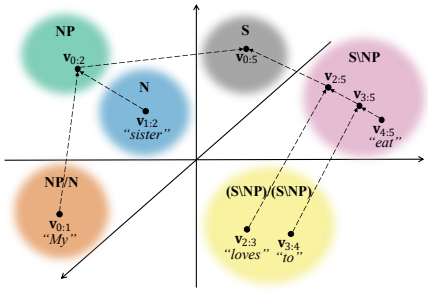

図 1　埋め込み空間上でのベクトルの再帰的な合成．

CCG では句構造の探索に先立って，各単語に尤もらしい CCG カテゴリ (Supertag) を割り当てる Supertagging という処理が行われることが一般的であり，これによって句構造の探索空間が大幅に削減され，探索効率が向上する [2]．ここで，既存の Supertagging 手法には LSTM，Transformer などのニューラルネットワークベースのエンコーダを用いることで各単語に割り当てられるカテゴリを分類問題として予測するモデル [3, 4] や，単語ごとに尤もらしいカテゴリを生成するモデル [5, 6, 7] が存在する．文の構成要素である単語や句の間に存在する依存関係や階層関係のモデリングの観点からみると，既存手法ではこれらの関係をニューラルネットワークを用いて非明示的にモデリングしているものが多い．

これに対し，本研究では Holographic Embeddings (HolE) [8] を用いて，図 1 に示すような埋め込み空間上でのベクトルの再帰的な合成を行うことで，Supertagging モデルに単語や句同士の依存関係を明示的に組み込む手法を提案する．また，ベクトルの再帰的合成によって得られる句の分散表現を活用した新たな構文解析アルゴリズムを導入し，依存関係の明示的なモデリングの CCG 構文解析における有用性について検証する．

2　Holographic Embeddings

Nickel らは知識グラフをベクトル空間に埋め込むための手法として Holographic Embeddings [8] を提案

した．知識グラフは複数の実体と，それらの関係性を表したエッジから構成されるグラフ構造のデータであり，知識グラフをベクトル空間上でモデリングするためには，各実体に付与された分散表現同士の相互作用を的確に捉えられるベクトルの合成演算子を設計することが重要となる．HolE ではこの合成演算子として巡回相関を採用しており，巡回相関はベクトル間の相互関係を捉えることに適している．

巡回相関の演算は $\star$ で表され，式 (1) で定義される．また式 (2) のように高速フーリエ変換 $\mathcal{F}$ およびその逆変換 $\mathcal{F}^{-1}$ を用いても計算可能であることが知られている．ここで，$\mathbf{c}_1, \mathbf{c}_2$ は合成元の d 次元ベクトル，$\mathbf{p}$ は合成後の d 次元ベクトル，$\odot$ はアダマール積，$\overline{\mathcal{F}(\cdot)}$ は複素共役をそれぞれ表す．

$$[\mathbf{p}]_k = [\mathbf{c}_1 \star \mathbf{c}_2]_k = \sum_{i=0}^{d-1} c_{1i} c_{2(k+i) \mod d} \quad (1)$$

$$\mathbf{p} = \mathbf{c}_1 \star \mathbf{c}_2 = \mathcal{F}^{-1}(\overline{\mathcal{F}(\mathbf{c}_1)} \odot \mathcal{F}(\mathbf{c}_2)) \quad (2)$$

単語と単語，単語と句，句と句など，自然言語の文を構成する要素間の階層関係・依存関係をモデリングするにあたり，巡回相関には以下のような数学的に好ましい性質が存在する．

- **非可換な演算である**：巡回相関では一般に $\mathbf{a} \star \mathbf{b} \neq \mathbf{b} \star \mathbf{a}$ が成り立つ．この性質は非対称な依存関係をモデリングするのに向いている．例えば，"right human" と "human right" という 2 つの名詞句に対して異なる分散表現を合成し，これらを区別することが可能である．
- **結合法則を満たさない**：巡回相関では一般に $(\mathbf{a} \star \mathbf{b}) \star \mathbf{c} \neq \mathbf{a} \star (\mathbf{b} \star \mathbf{c})$ が成り立つ．この性質は句構造解析が明らかにしようとする自然言語の階層関係をモデリングするのに重要である．例えば，*"saw a girl with a telescope"* という句には少なくとも 2 つの階層関係・内部構造が考えられる：*"((saw (a girl)) (with (a telescope)))"*，*"(saw ((a girl) (with (a telescope))))"*．このような単語列に対して，分散表現の合成演算が結合法則を満たさない場合，複数存在しうる階層関係のそれぞれを，異なる分散表現として区別することが可能となる．

以上より，我々は HolE ならびに巡回相関を CCG 構文解析において文中の単語や句の間に存在する依存関係をモデリングするための手法として採用する．

3 Holographic CCG

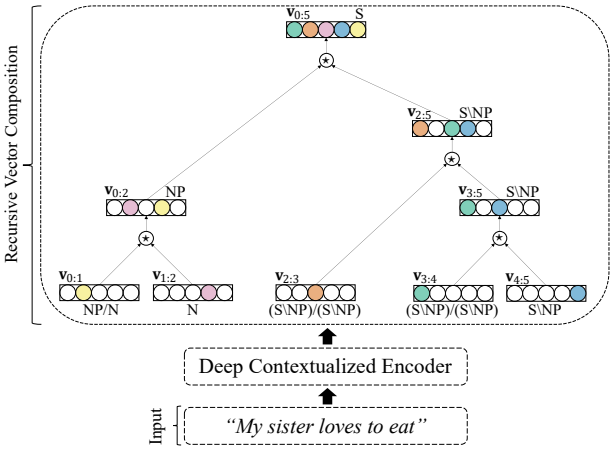

図 2　単語分散表現の再帰的合成．

3.1 単語分散表現の再帰的合成

ここでは，CCG の Supertagging モデルにおいて，文の構成要素間に存在する依存関係を明示的にモデリングするための手法として，巡回相関による単語分散表現の再帰的合成を提案する．本手法では文中における特定の句を構成している複数の単語の分散表現を巡回相関によって再帰的に合成することで，句全体としての分散表現を計算する．単語分散表現の再帰的合成の概略図を図 2 に示し，図中の具体例を用いて本手法の概要を説明する．まず，入力文 (*"My sister loves to eat"*) を事前学習済みのエンコーダに入力することで各単語のベクトル $(\mathbf{v}_{0:1}, \mathbf{v}_{1:2}, \cdots, \mathbf{v}_{4:5})$ を得る．ベクトルの添字 $i:j$ は，入力文中の $i+1$ 番目の単語から j 番目の単語が含まれる区間 $[i, j]$ を意味する．なお，全ての実験においてエンコーダには RoBERTa-large [9] を採用しており，各単語は 1024 次元のベクトルに変換される．次に各単語のベクトルを CCG の組み合わせ規則に従って再帰的に合成することで，文を構成する各句の分散表現を計算する．例えば，図中の *"loves to eat"* という動詞句に対応するベクトル $\mathbf{v}_{2:5}$ は，以下の通りにベクトルを再帰的に合成することで計算される．

$$\mathbf{v}_{2:5} = \mathbf{v}_{2:3} \star \mathbf{v}_{3:5} = \mathbf{v}_{2:3} \star (\mathbf{v}_{3:4} \star \mathbf{v}_{4:5})$$

同様の計算を，文全体としてのベクトル $\mathbf{v}_{0:5}$ が得られるまで繰り返し行う．

$$\begin{aligned}\mathbf{v}_{0:5} &= \mathbf{v}_{0:2} \star \mathbf{v}_{2:5} \\ &= (\mathbf{v}_{0:1} \star \mathbf{v}_{1:2}) \star (\mathbf{v}_{2:3} \star \mathbf{v}_{3:5}) \\ &= (\mathbf{v}_{0:1} \star \mathbf{v}_{1:2}) \star (\mathbf{v}_{2:3} \star (\mathbf{v}_{3:4} \star \mathbf{v}_{4:5}))\end{aligned}$$

なお，巡回相関を再帰的に適用すると合成後のベクトルのノルムが急激に増大し，数値計算上好ましくないため，巡回相関によって合成したベクトルに対して，式 (3) で表されるノルム制約を加えている．

$$\mathbf{p}' = \begin{cases} k\mathbf{p}/(\|\mathbf{p}\| + \epsilon) & \text{if } \|\mathbf{p}\| \geq k \\ \mathbf{p} & \text{otherwise} \end{cases} \tag{3}$$

ただし，$\mathbf{p}, \mathbf{p}'$ はそれぞれノルム制約の適用前と適用後のベクトルを表しており，k はノルムの最大値である．本研究では，全ての実験において $k = 10^3, \epsilon = 10^{-12}$ に設定した．

3.2 Supertagging

入力文を構成する全ての単語と句に関して，それらのベクトル $\mathbf{v}_{i:i+1}, \mathbf{v}_{i:j}$ を計算した後，これらを順伝播型ニューラルネットワークに入力することで，単語と句それぞれに関するカテゴリ割当の確率分布 $P_w(i, i+1), P_p(i, j)$ と，区間 $[i, j]$ において句が成立する確率 $P_s(i, j)$ をそれぞれ計算する．

$$P_w(i, i+1) = SM(\mathbf{Q}_w \sigma(LN(\mathbf{U}_w \mathbf{v}_{i:i+1} + \mathbf{b}_w)) + \mathbf{c}_w)$$
$$P_p(i, j) = SM(\mathbf{Q}_p \sigma(LN(\mathbf{U}_p \mathbf{v}_{i:j} + \mathbf{b}_p)) + \mathbf{c}_p)$$
$$P_s(i, j) = Sigmoid(\mathbf{q}_s^\top \sigma(LN(\mathbf{U}_s \mathbf{v}_{i:j} + \mathbf{b}_s)) + \mathbf{c}_s)$$

ここで，$\mathbf{Q}_w, \mathbf{Q}_p, \mathbf{q}_s^\top, \mathbf{U}_w, \mathbf{U}_p, \mathbf{U}_s, \mathbf{b}_w, \mathbf{b}_p, \mathbf{b}_s, \mathbf{c}_w, \mathbf{c}_p, \mathbf{c}_s$ はモデルの学習対象となる重み行列およびベクトルであり，$\sigma(\cdot)$ は ReLU による非線形変換，$LN(\cdot)$ は Layer Normalization，$SM(\cdot)$ はソフトマックス関数，$Sigmoid(\cdot)$ はシグモイド関数をそれぞれ表す．また，全ての順伝播型ニューラルネットワークにおいて，ReLU による非線形変換の直後に Dropout 層を挿入している．これらの単語・句に対するカテゴリ割当の確率分布および句の成立確率が正解に合致するようにモデルを学習させることで，単語や句の分散表現が相互に影響を及ぼし合い，結果として構成要素間の依存関係を明示的に与えた状況下での Supertagging モデルの学習が実現される．

モデルの学習には誤差逆伝播法を用いる．まず，単語のカテゴリ割当 $P_w(i, i+1)$ の予測誤差 $\mathcal{L}_w$ と句のカテゴリ割当 $P_p(i, j)$ の予測誤差 $\mathcal{L}_p$ をそれぞれ交差エントロピー誤差を基準として計算する．また，これらに加えて句の成立確率 $P_s(i, j)$ の予測誤差 $\mathcal{L}_s$ を，二値交差エントロピー誤差を基準として計算する．そして，これらの誤差の一部または全部を逆伝播することでモデルパラメータの最適化を行う．例えば，3 つの誤差全てを逆伝播する場合，合計誤差の計算およびモデルパラメータの更新は式 (4)，(5) でそれぞれ表される．

$$\mathcal{L} = \mathcal{L}_w + \mathcal{L}_p + \mathcal{L}_s \tag{4}$$
$$\theta \leftarrow \theta - \mu \frac{\partial \mathcal{L}}{\partial \theta} \tag{5}$$

ここで，$\mathcal{L}$ は逆伝播される合計誤差，θ はモデルパラメータ，μ は学習率をそれぞれ表す．

モデルの学習終了後，開発用データ及びテスト用データに対して Supertagging を行う際には，まず入力文中の各単語をエンコーダを通して分散表現に変換する．そして，これらのベクトルを順伝播型ニューラルネットワークに入力することで $P_w(i, i+1)$ を計算し，各単語に対するカテゴリ割当の予測を行う．ここで，各単語に対する割当確率が 0.1 以上となるカテゴリ全てを各単語の Supertag として付与する．

3.3 Span-based Parsing

提案手法によって計算される句の分散表現は，句に対するカテゴリ割当を直接予測しながら句構造の探索を行う Span-based の構文解析アルゴリズム (以下 Span-based Parsing) に適用可能である．Span-based Parsing は Stern らが PCFG による句構造解析に導入し [10]，Clark が同様の手法を CCG に適用することで大幅な性能向上を達成した [11]．Span-based Parsing では，単語の分散表現から文中の特定の区間・句に対応する分散表現を計算し，その分散表現を用いて区間が句を形成する確率および，句に対して割り当てられるカテゴリの予測を行う．我々は，CKY アルゴリズムに対して，提案手法における句のカテゴリ割当の確率分布 $P_p(i, j)$ と句の成立確率 $P_s(i, j)$ の情報を組み込むことによって，新たな Span-based Parsing アルゴリズムを導入した．本アルゴリズムの詳細については付録 A に示す．

4 実験

4.1 データセット

モデルの学習・評価のためのデータセットには，CCG で一般的に用いられる英語 CCGbank [12] を使用した．CCGbank は Penn Treebank と同じ Wall Street Journal の記事に対して CCG のカテゴリがアノテーションされたデータセットであり，24 個 (00-23) のセクションで構成されている．本実験では既存研究のデータセット分割方法に従い，セクション 02-21

表 1　CCGbank の統計情報.

	Train	Dev	Test
Section number	02-21	00	23
Number of sentences	39,604	1,913	2,407
Number of words	929,552	45,422	55,371

表 2　モデルの学習に関するハイパーパラメータ.

Hyperparameters	**Values**
Epochs	10
Batch size	16
Learning rates	1e-4(base), 1e-5(fine-tune)
AdamW $\beta's$	0.9, 0.999
AdamW ϵ	1e-6
Weight decay	0.01
Dropout probability	0.2

を学習用データ，セクション 00 を開発用データ，セクション 23 をテスト用データとして用いた．データセットに関する統計情報を表 1 に示す．

4.2　実験条件

モデルの学習に関する各種ハイパーパラメータを表 2 に示す．Optimizer には AdamW [13] を使用し，学習率は RoBERTa-large のファインチューニング部分 (fine-tune) とその他の部分 (base) で異なる値を設定した．

学習終了後，提案モデルによって Supertagging がなされた文を事前学習済みの CCG 構文解析モデルである C&C Parser [14] の入力とすることで構文解析を行った．また，同様に我々が導入した Span-based Parsing による構文解析も行った．

提案手法の有効性を定量的に評価するにあたり，異なる誤差の組み合わせを用いてモデルの学習を行った．まず単語レベルでのカテゴリ割当の誤差 $\mathscr{L}_w$ のみを学習に用いるモデルをベースラインモデルとし，これに対して句の分散表現から計算される誤差 $\mathscr{L}_p, \mathscr{L}_s$ の一方，もしくは両方を合計した誤差を用いるものを提案モデルとする．各モデルを異なる乱数シード値から 8 回ずつ学習させ，ベースラインモデルと提案モデルの各種評価指標の値を比較した．

4.3　評価指標

既存研究に従い，モデルの評価指標には Supertagging の精度 (**Acc**) と構文解析結果から得られる単語間依存関係の情報として Labeled F-score (**LF**) を用いた．

表 3　開発用データにおける提案法の性能.

Objective	**Parser**	**Acc**	**LF**
$\mathscr{L}_w$(baseline)	C&C	96.46±0.02	91.86±0.03
$\mathscr{L}_w + \mathscr{L}_p$	C&C	96.56±0.01	91.95±0.02
$\mathscr{L}_w + \mathscr{L}_s$	C&C	96.51±0.03	91.89±0.03
$\mathscr{L}_w + \mathscr{L}_p + \mathscr{L}_s$	C&C	**96.58±0.02**	92.00±0.03
	Span	–	**92.47±0.07**

表 4　テストデータにおける既存手法との性能比較.

Model	**Parser**	**Acc**	**LF**
Vaswani et al. [3]	C&C	94.24	88.32
Tian et al. [4]	C&C	96.25	90.58
Bhargava and Penn [5]	C&C	96.00	90.9
Prange et al. [6]	C&C	96.22	90.91
Liu et al. [7]	C&C	96.05	89.80
Clark [11] (Transformer を使用)	Span	–	**92.9**
提案モデル ($\mathscr{L}_w + \mathscr{L}_p + \mathscr{L}_s$)	C&C	**96.50**	**92.11**
	Span	–	92.60

5　結果

ベースライン手法と提案モデルそれぞれの開発用データにおける性能を表 3 に示す．有意水準 1%で片側 t 検定を行った結果，$\mathscr{L}_w + \mathscr{L}_p + \mathscr{L}_s$ を学習に用いた提案モデルはベースラインモデルを Supertagging 精度と C&C Parser における Labeled F-score の両者において有意に上回っていることが確認された．さらに，Span-based Parsing は C&C Parser をさらに上回る性能を発揮している．

次に，テスト用データにおける既存手法との比較を表 4 に示す．提案モデルの Supertagging 精度と C&C Parser における Labeled F-score は全ての既存手法を上回っており，Span-based Parsing における Labeled F-score も，現状の最高性能モデルである Clark [11] に迫る性能を発揮している．

6　おわりに

本研究では，CCG の Supertagging モデルにおいて文の構成要素間の依存関係を明示的にモデリングする手法を提案し，また，提案手法から計算される句の分散表現を活用した Span-based Parsing アルゴリズムを導入した．提案手法はベースライン手法を上回る最高性能を発揮し，依存関係の明示的なモデリングが CCG 構文解析にとって有効であることを示した．

本研究の今後の展開としては，現状の教師あり学習手法から教師なし学習手法への拡張や，ロボットのセンサー系から得られるマルチモーダル情報と言語情報の統合への応用などを予定している．

謝辞

本研究は，JST，ムーンショット型研究開発事業，JPMJMS2033 の支援を受けたものです．

参考文献

[1] Mark Steedman. **The Syntactic Process**, Vol. 24. MIT press Cambridge, MA, 2000.

[2] Stephen Clark and James R Curran. The Importance of Supertagging for Wide-Coverage CCG Parsing. In **COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics**, pp. 282–288, 2004.

[3] Ashish Vaswani, Yonatan Bisk, Kenji Sagae, and Ryan Musa. Supertagging with LSTMs. In **Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 232–237, 2016.

[4] Yuanhe Tian, Yan Song, and Fei Xia. Supertagging Combinatory Categorial Grammar with Attentive Graph Convolutional Networks. **arXiv preprint arXiv:2010.06115**, 2020.

[5] Aditya Bhargava and Gerald Penn. Supertagging with CCG primitives. In **Proceedings of the 5th Workshop on Representation Learning for NLP**, pp. 194–204, 2020.

[6] Jakob Prange, Nathan Schneider, and Vivek Srikumar. Supertagging the Long Tail with Tree-Structured Decoding of Complex Categories. **Transactions of the Association for Computational Linguistics**, Vol. 9, pp. 243–260, 2021.

[7] Yufang Liu, Tao Ji, Yuanbin Wu, and Man Lan. Generating CCG Categories. **Proceedings of the AAAI Conference on Artificial Intelligence**, Vol. 35, No. 15, pp. 13443–13451, May 2021.

[8] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. Holographic Embeddings of Knowledge Graphs. In **Proceedings of the AAAI Conference on Artificial Intelligence**, Vol. 30, 2016.

[9] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. **arXiv preprint arXiv:1907.11692**, 2019.

[10] Mitchell Stern, Jacob Andreas, and Dan Klein. A Minimal Span-Based Neural Constituency Parser. **arXiv preprint arXiv:1705.03919**, 2017.

[11] Stephen Clark. Something Old, Something New: Grammar-based CCG Parsing with Transformer Models. **arXiv preprint arXiv:2109.10044**, 2021.

[12] Julia Hockenmaier and Mark Steedman. CCGbank: a corpus of CCG derivations and dependency structures extracted from the Penn Treebank. **Computational Linguistics**, Vol. 33, No. 3, pp. 355–396, 2007.

[13] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In **International Conference on Learning Representations**, 2019.

[14] Stephen Clark, Darren Foong, Luana Bulat, and Wenduan Xu. The Java Version of the C&C Parser: Version 0.95. **Technical report, University of Cambridge Computer Laboratory, August**, 2015.

A Span-based Parsing アルゴリズム

Algorithm 1 Span-based CKY Parsing

$\mathbf{v}_{0:1}, \mathbf{v}_{1:2}, \cdots, \mathbf{v}_{n-1:n} = Encode(w_1, w_2, \cdots, w_n)$ ▷ エンコーダで文中の各単語をベクトルに変換
for $i = 0, \cdots, n-1$ **do**
  $P_w(i, i+1) = SM(\mathbf{Q}_w \sigma(LN(\mathbf{U}_w \mathbf{v}_{i:i+1} + \mathbf{b}_w)) + \mathbf{c}_w)$ ▷ 単語に対するカテゴリ割当確率を計算
  for $C \in \{X | P_w(i, i+1)[X] > t_w = 0.1\}$ **do** ▷ $P_*(i, j)[X]$ は区間 $[i, j]$ に対するカテゴリ X の割当確率
    $prob[i, i+1, C] = \log P_w(i, i+1)[C]$
    $vector[i, i+1, C] = \mathbf{v}_{i:i+1}$
for $l = 2, \cdots, n$ **do**
  for $i = 0, \cdots, n-l$ **do**
    $j = i + l$
    for $k = i+1, \cdots, j-1$ **do**
      for $C_1 \in \{X | prob[i, k, X] > 0\}$ **do**
        $\mathbf{v}_{i:k} = vector[i, k, C_1]$
        for $C_2 \in \{X | prob[k, j, X] > 0\}$ **do**
          $\mathbf{v}_{k:j} = vector[k, j, C_2]$
          $\mathbf{v}_{i:j} = \mathbf{v}_{i:k} \star \mathbf{v}_{k:j}$ ▷ 巡回相関でベクトルを合成
          $P_s(i, j) = Sigmoid(\mathbf{q}_s^\top \sigma(LN(\mathbf{U}_s \mathbf{v}_{i:j} + \mathbf{b}_s)) + \mathbf{c}_s)$ ▷ 句が成立する確率を計算
          if $P_s(i, j) > t_s = 0.01$ **then**
            $P_p(i, j) = SM(\mathbf{Q}_p \sigma(LN(\mathbf{U}_p \mathbf{v}_{i:j} + \mathbf{b}_p)) + \mathbf{c}_p)$ ▷ 句に対するカテゴリ割当確率を計算
            for $C \in \{X | C_1 C_2 \rightarrow X \in R\}$ **do** ▷ R は組み合わせ規則
              if $P_p(i, j)[C] > t_p = 0.01$ **then**
                $p = \log P_s(i, j) + \log P_p(i, j)[C] + prob[i, k, C_1] + prob[k, j, C_2]$
                if $p > prob[i, j, C]$ **then**
                  $prob[i, j, C] = p$
                  $backpointer[i, j, C] = (k, C_1, C_2)$
                  $vector[i, j, C] = \mathbf{v}_{i:j}$

本研究において我々が導入した Span-based Parsing のアルゴリズムを Algorithm 1 に示す．簡略化のため，上記アルゴリズムでは，CCG の組み合わせ規則のうち，Binary の規則のみを前提とした場合の処理手順を示している．実際，CCG には Type-raising などの Unary の規則が存在するため，上記アルゴリズムに加えて，Unary の規則によって生成されるカテゴリをチャートに追加する必要がある．