5E-03

量子アニーリングによる構文解析手法*

小見山　朋子†　　鈴木　智博‡
山梨大学大学院　医工農学総合教育部　工学専攻

1　はじめに

近年, 量子アニーリングの動作を再現した, 量子アニーリングマシンの開発が進められている. 量子アニーリングは, 2 次式で表されるエネルギー関数を最小化する解を求める. 現状の量子アニーリングマシンでは利用できる量子ビット数が限られているため, 2 次式かつ量子ビット数が少なくなるような問題の定式化が必要である. 本稿では, 量子アニーリングの適用例が少ない自然言語処理, 特に, 構文解析に量子アニーリングを適用する手法を示す.

2　構文解析

自然言語処理における文の理解は,

形態素解析 → 構文解析 → 意味解析 → 文脈解析

の流れで行われる. 構文解析は, 文が与えられた際に, 文を構成する語の文法的役割を決定する処理のことである. 構文解析の上流工程である形態素解析により得られた品詞列に対して, 文法規則の中から適切な文法を適用する [1, 2].

構文解析の手法として, Dependency Parsing と Consituency Parsing がある. Dependency Parsing は, 解析対象である文中の単語の依存関係に基づいた構文解析手法であり, 構文解析の下流工程である意味解析がしやすいといった利点がある. 一方, Constituency Parsing は, 名詞句や動詞句といった構成要素に基づいた構文解析手法で, 文脈自由文法のような生成規則を逆向きに適用していくことで構文木を生成する [3, 4]. どちらの構文解析手法も大量の文法規則もしくは依存関係の中から必要な規則だけを探して適用するため, 厳密に行うと非常に時間がかかる. また, 意味に関わらず文法規則だけで解析を行うため, 曖昧性が高くなると組み合わせが増加する. そのため, 構文解析では, 常に組合せ爆発が起こる可能性があり, 全ての可能性を解析するのは非常に困難である [1]. 現状の実用的なシステムでは, 枝刈り等を行うことで解析を打ち切っている.

*Syntactic parsing in natural language processing using quantum annealing
†Tomoko Komiyama, University of Yamanashi
‡Tomohiro Suzuki, University of Yamanashi

3　提案手法:　有効グラフによる Constituency Parsing

本手法では, 文法規則を品詞を頂点とし, 図 1 のように文法規則にしたがって頂点間を結ぶ辺を設定することで有向グラフを生成し, これを用いて構文解析を行う. 構文解析は, 形態素解析により得られた品詞である

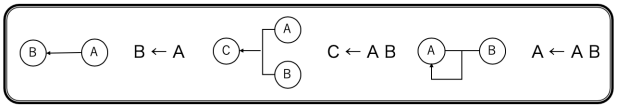

図 1: 文法規則から有向グラフへの変換

始点を全て通り, 「文」である終点にたどり着く経路を探索することで行う.

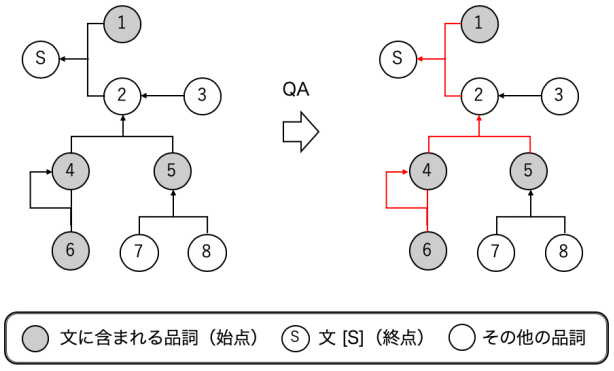

図 2: 有向グラフによる Constituency Parsing

本手法の経路探索における制約条件を (2), (3), (4) 式で定式化し, 条件を満たしたときに最小エネルギーとなる関数

$$H = H_{start} + H_{goal} + H_{other} \tag{1}$$

を定義した. 以下, 変数 $x_{ij} \in \{0, 1\}$ は頂点 i から j への有向辺を経路に含めるかどうか, S は始点集合, V は頂点集合, g は終点を表す.
(1) 始点:　（流入辺 − 流出辺）= 1

$$H_{start} = \sum_{s \in S} \left\{ \sum_{j \in V} (x_{sj} - x_{js}) - 1 \right\}^2 \tag{2}$$

(2) 終点: （流出辺 − 流入辺）= 1

$$H_{goal} = \left\{ \sum_{j \in V} (x_{gj} - x_{jg}) - 1 \right\}^2 \quad (3)$$

(3) その他の頂点: （流入辺 − 流出辺）= 0

$$H_{other} = \sum_{j \in V} \left\{ \sum_{j \in V} (x_{ij} - x_{ji}) - 1 \right\}^2 \quad (4)$$

4 評価実験

GPU ベースのアニーリングマシンである Fixstars Amplify Annealing Engine (Amplify AE)[6] を用いて, 提案手法の評価を行う. 本実験では, Amplify AE のアニーリング時間は 1 秒, その他設定可能なパラメータは全てデフォルトの値に設定する.

4.1 実験 1: 曖昧性がある文に対する構文解析

はじめに, 「Time flies like an arrow」という品詞の曖昧性を含む文を用いて, 提案手法による構文解析を行う. 各単語の品詞と文法規則は表 1 を適用し, 実験回数は 1000 回とした. 実験結果を図 3 に示す. 3 種類の構

表 1: 各単語の品詞と文法規則

N(noun) = {Times, flies, atrrow}, D(det) = {an}
V(verb) = {flies, like}, P(prep) = {like}
S ← {NP VP}, NP ← {N |D N |NP PP}
VP ← {V VP |V NP |V PP}, PP ← {P NP}

文解析が得られ, 平均実行時間は 1.40 秒となった. この結果から, とり得る構文解析結果が全て得られ, また, それぞれの構文解析が得られる確率は, (a) が 40.6%, (b) が 34.1%, (c) が 25.3% と偏りが出ることがわかった. ここから, 提案手法による構文解析が可能であると考える. 解の偏りが出る原因は, 量子アニーリングの特有の性質によるものであると考える. また, どの構文解析が適切であるかの判断は構文解析の下流工程である意味解析で行われるため今回は議論しない.

4.2 実験 2: 文法規則が多い場合の構文解析

次に, 正しい構文解析結果がわかる 4 文（6〜11word）を用いて, 提案手法により得られる解の評価を行う. 文法規則は 20 文の正しい構文解析結果から生成し, 実験回数は 800（200 × 4 文）回である. また, 提案手法では図 4 のような文法的に意味が等価であるが, 文法規則を冗長にとる解が得られることがある. このような文法的な意味が等価であるものは区別せずに評価を行なった. 実験結果から, 4.25%で提案手法により正解が得られることがわかった. 得られた解を解析したところ, 本手法では単語の語順を反映できないことが正解の割合が低くなった原因であることがわかった.

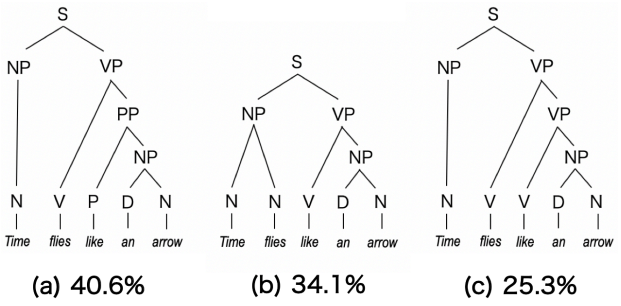

図 3: 実験 1 の結果

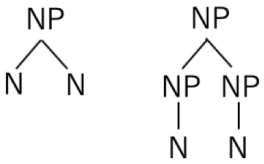

図 4: 冗長な構文解析

5 おわりに

本稿では, 量子アニーリングによる構文解析手法である有向グラフによる Constituency Parsing の提案と評価実験の結果を示した. 提案手法により構文解析は可能であるが, 単語の語順を見ないため, 文法規則数が増えると正しい解析結果が得られないことがわかった. 今後は, 単語の語順を見れるような定式化の検討を行うことで, より現実的な問題を用いた構文解析ができるように改良を行なっていく.

参考文献

[1] 天野真家, 石崎俊, 宇都呂武仁, 成田真澄, 福本淳一, “自然言語処理”, オーム社, 2007.

[2] 奥村学, 鶴岡慶雅, 宮尾祐介, ”構文解析”, コロナ社, 2017.

[3] 黒橋禎夫, 柴田知秀, “自然言語処理概論”, サイエンス社, 2016.

[4] Daniel Jurafsky, James H. Martin, “Speech and Language Processing”.

[5] A. Lucas, ”Ising formulations of many NP problems”, Frontiers in Physics, 2014.

[6] Fixstars Amplify Annealing Engine, https://amplify.fixstars.com